Session 1

25-May-2021

09:00am – 10:15pm

Sections 1, 3, 4, 5, 6

QA1

Part (a) [2 marks]

Consider the following declarations:

```
float x = 15.0, y = 6.0;
int a = 4, b = 3;
float z;
```

What is the value of z computed by the following assignment?

```
z = x / y + a / b ;
```

Ans: 3.500000

Part (b) [8 marks]

Let A[] be an unsorted array of n positive integers. We want to print the elements of A[] in the ascending order. Assume that the elements of A[] are distinct from one another. Complete the following code by filling in the blanks. In the variable lastprinted, we store the array element that is printed last so far. You are not permitted to use any other variables.

```
      #define MINUS_INFINITY -1000000000

[A]   void listall ( __________ )  /* Pass A and n */
      {
         int i, j;
         int lastprinted, minidx;

         lastprinted = MINUS_INFINITY;
         for (i=0; i<n; ++i) {
            /* Find in the variable minidx the index of the smallest element larger than the one last printed */
            minidx = -1 ;
[B]         for ( __________ ) { /* Loop over j */
[C]            if ( __________ ) minidx = j;
            }
            printf("%d ", A[minidx]);
[D]         __________
         }
         printf("\n");
      }
```

Ans:

[A] `int A[], int n`
[B] `j=0; j<n; ++j`
[C] `(A[j] > lastprinted) && ((minidx == -1) || (A[j] < A[minidx]))`
[D] `lastprinted = A[minidx];`

QA2

Part (a) [2 marks]

Consider the following declarations:

```
float x = 18.0, y = 4.0;
int a = 4, b = 3;
float z;
```

What is the value of z computed by the following assignment?

```
z = x / y + a / b ;
```

Ans: 5.500000

Part (b) [8 marks]

Let A[] be an unsorted array of n positive integers. We want to print the elements of A[] in the descending order. Assume that the elements of A[] are distinct from one another. Complete the following code by filling in the blanks. In the variable lastprinted, we store the array element that is printed last so far. You are not permitted to use any other variables.

```
      #define PLUS_INFINITY 1000000000
      /* Assume that all the elements of A are smaller than PLUS_INFINITY */

[A]   void listall ( __________ )  /* Pass A and n */
      {
         int i, j;
         int lastprinted, maxidx;

         lastprinted = PLUS_INFINITY;
         for (i=0; i<n; ++i) {
            /* Find in the variable maxidx the index of the largest element smaller than the one last printed */
            maxidx = -1 ;
[B]         for ( __________ ) { /* Loop over j */
[C]            if ( __________ ) maxidx = j;
            }
            printf("%d ", A[maxidx]);
[D]         __________
         }
         printf("\n");
      }
```

Ans:

[A] `int A[], int n`
[B] `j=0; j<n; ++j`
[C] `(A[j] < lastprinted) && ((maxidx == -1) || (A[j] > A[maxidx]))`
[D] `lastprinted = A[maxidx];`

QA3

Part (a) [2 marks]

Consider the following declarations:

```
float x = 5.0;
int a = 7, b = 2;
float z;
```

What is the value of z computed by the following assignment?

```
z = a / b * a / b * x ;
```

Ans: 50.000000

Part (b) [8 marks]

Let A[] be an unsorted array of n positive integers. We want to print the elements of A[] in the ascending order. Assume that the elements of A[] are distinct from one another. After we print an element of A[], we replace that element by –1 so that this element is not considered again for minimum calculations in future. Complete the following code by filling in the blanks. You are not permitted to use any other variables.

```
[A]   void listall ( __________ )  /* Pass A and n */
      {
         int i, j;
         int minidx;

         for (i=0; i<n; ++i) {
            /* Find in the variable minidx the index of the smallest unprinted element */
            minidx = -1 ;
[B]         for ( __________ ) { /* Loop on j */
[C]            if ( __________ ) minidx = j;
            }
            printf("%d ", A[minidx]);
[D]         __________
         }
         printf("\n");
      }
```

Ans:

[A] `int A[], int n`
[B] `j=0; j<n; ++j`
[C] `(A[j] >= 0) && ((minidx == -1) || (A[j] < A[minidx]))`
[D] `A[minidx] = -1;`

QA4

Part (a) [2 marks]

Consider the following declarations:

```
float x = 5.0;
int a = 8, b = 3;
float z;
```

What is the value of z computed by the following assignment?

```
z = a / b * a / b * x ;
```

Ans: 25.000000

Part (b) [8 marks]

Let A[] be an unsorted array of n positive integers. We want to print the elements of A[] in the descending order. Assume that the elements of A[] are distinct from one another. After we print an element of A[], we replace that element by –1 so that this element is not considered again for maximum calculations in future. Complete the following code by filling in the blanks. You are not permitted to use any other variables.

```
[A]   void listall ( __________ )  /* Pass A and n */
      {
         int i, j;
         int maxidx;

         for (i=0; i<n; ++i) {
            /* Find in the variable maxidx the index of the largest unprinted element */
            maxidx = -1 ;
[B]         for ( __________ ) { /* Loop on j */
[C]            if ( __________ ) maxidx = j;
            }
            printf("%d ", A[maxidx]);
[D]         __________
         }
         printf("\n");
      }
```

Ans

[A] `int A[], int n`
[B] `j=0; j<n; ++j`
[C] `(A[j] >= 0) && ((maxidx == -1) || (A[j] > A[maxidx]))`
[D] `A[maxidx] = -1;`

QB1

Part (a) [2 marks]

What will be printed by the following program fragment if x is an int type variable containing the value 7361, and y is an int type variable?

```
do {
   y=x%10;
   printf("%c", '0' + y);
} while((x = x/10) != 0);
```

Ans: 1637

Part (b) [8 marks]

Fill in the blanks to complete the following program that reads a string S as input (assume that the string can be at most 50 characters long and will contain only alphabetic letters (lower case and upper case)), and forms a new string R that contains, in lower case and in order, all letters (from the alphabet) that are not present in S, in either lower case or upper case, and finally, prints R. For example, if S = "rtaBfFHgDceAPQ", then R = "ijklmnosuvwxyz". You cannot use any string functions or ascii code values of any character.

```
int main()
{
    char S[100], R[100], temp;
    int i, j, X[26];

    scanf("%s", S);

    /* X will keep track if a letter is present in S */
    for(i = 0; i < 26; i++) X[i] = 0;

     /* Loop over S to search for letters and update X */
[A] for ( __________ ) {
        if (S[i] >= 'A' && S[i] <= 'Z')
[B]         temp = __________ ;
        else
            temp = S[i];
[C]     X[ __________ ] = 1;
    }

     /* Write to R */
    j = 0;
    for ( i = 0; i<26; i++ )
[D]     if (X[i] == 0) { __________ }

    R[j] = '\0';

    printf("R = %s\n", R);
}
```

Ans:

[A] `i = 0; S[i] != '\0'; i++`
[B] `'a' + S[i] - 'A'`
[C] `temp - 'a'`
[D] `R[j++] = 'a' + i;`

QB2

Part (a) [2 marks]

What will be printed by the following program fragment if x is an int type variable containing the value 3051, and y is an int type variable?

```
do {
   y=x%10;
   printf("%c", 'A' + y);
} while((x = x/10) != 0);
```

Ans: BFAD

Part (b) [8 marks]

Fill in the blanks to complete the following program that reads a string S as input (assume that the string can be at most 100 characters long and can contain any character), and forms a new string R that contains all digits (in order) that are not present in S, and finally, prints R. For example, if S = "dB@dae21#$a5", then R = "0346789". You cannot use any string functions or ascii code values of any character.

```
int main()
{
    char S[101], R[101];
    int i, j;
    int Y[10];

    scanf("%s", S);

    /* Y will keep track if a digit is present in S */
    for(i = 0; i < 10; i++) Y[i] = 0;

    /* Loop over S to search for digits and update Y */
[A] for ( __________ ) {
        /*Ignore non-digits */
[B]     if ( __________ ) continue;

[C]     Y[ __________ ] = 1;
    }

     /* Write to R */
    j = 0;
    for ( i = 0; i<10; i++ )
[D]     if (Y[i] == 0) { __________ }

    R[j] = '\0';

    printf("R = %s\n", R);

}
```

Ans:

[A] `i = 0; S[i] != '\0'; i++`
[B] `S[i] < '0' || S[i] > '9'`
[C] `S[i] - '0'`
[D] `R[j++] = '0' + i;`

QB3

Part (a) [2 marks]

What will be printed by the following program fragment if x is an int type variable containing the value 7235, and y is an int type variable?

```
do {
   y = x % 10;
   printf("%c", '9' - y);
} while ((x = x/10) != 0);
```

Ans: 4672

Part (b) [8 marks]

Fill in the blanks below so that the program does the following: reads a string S as input (assume that the string can be at most 50 characters long and can contain any characters), and forms a new string R that contains, in upper case and in order, all the letters (from the alphabet) that are not present in lower case in S, and finally, prints R. For example, if S = "xRt$aA%3*2d(BaZk", then R = "BCEFGHIJLMNOPQRSUVWYZ". You cannot use any string functions or ascii code values of any character.

```
      int main()
      {
          char S[100], R[100];
          int i, j, Z[26];

          scanf("%s", S);

          /* Z will keep track if a letter is present in S */
          for(i = 0; i < 26; i++) Z[i] = 0;

          /* Loop over S to search for lower-case letters and update Z */
[A]       for ( __________ ) {
              /* Ignore all characters other than lower-case letters */
[B]           if ( __________ ) continue;

[C]           Z[ __________ ] = 1;
          }

           /* Write to R */
          j = 0;
          for ( i = 0; i<26; i++ )
[D]           if (Z[i] == 0) { __________ }

          R[j] = '\0';
          printf("R = %s\n", R);
      }
```

Ans:

[A] `j = 0; S[j] != '\0'; j++`
[B] `S[j] < 'a' || S[j] > 'z'`
[C] `S[j] - 'a'`
[D] `R[j++] = 'A' + i;`

QC1

Part (a) [4 marks]

Express the return value f(n) as a mathematical function of n. Assume that n is a positive integer.

```
int f ( int n )
{
   int a, b, s;

   s = a = b = 0;
   while ((a < n) || (b < n)) {
      if (a <= b) { s += b; ++a; }
      else { s += a; ++b; }
   }
   return s;
}
```

Ans: n^2

Part (b) [6 marks]

Write a function find_power() which takes an integer a as argument and returns the largest positive integer k for which $2^k \leqslant a$. Assume that $a \geqslant 2$. Do not use any math library function. You do not need to write any main() function.

Ans:

```
int find_power(int a) {
    int i;              //"i" is the loop index
    int power = 1;      //"power" variable contains the i^th power of a

    for(i = 1; ;i++){
        power = power * 2;
        if(power > a){
            return (i-1);
        }
    }
}
```

QC2

Part (a) [4 marks]

Express the return value f(n) as a mathematical function of n. Assume that n is a positive integer.

```
int f ( int n )
{
   int a, b, s;

   s = a = b = 0;
   while ((a < n) || (b < n)) {
      if (a <= b) { s += a; ++a; }
      else { s += b; ++b; }
   }
   return s;
}
```

Ans: $n^2 - n$

Part (b) [6 marks]

Write a function find_power() which takes an integer a as argument and returns the smallest positive integer k for which $2^k \geqslant a$. Assume that $a \geqslant 2$. Do not use any math library function. You do not need to write any main() function.

Ans:

```
int find_power(int a) {
    int i;              //"i" is the loop index
    int power = 1;      //"power" variable contains the i^th power of 2
    for(i = 1; ;i++){
        power = power * 2;
        if(a <= power){
            return (i);
        }
    }
}
```

QC3

Part (a) [4 marks]

Express the return value f(n) as a mathematical function of n. Assume that n is a positive integer.

```
int f ( int n )
{
   int a, b, s;

   s = a = b = 0;
   while ((a < n) || (b < n)) {
      if (a <= b) ++a; else ++b;
      s += a;
   }
   return s;
}
```

Ans: $n^2 + n$

Part (b) [6 marks]

Write a function find_power() which takes an integer a as argument and returns the largest positive integer k for which $3^k \leqslant a$. Assume that $a \geqslant 3$. Do not use any math library function. You do not need to write any main() function.

Ans:

```
int find_power(int a) {
    int i;              //"i" is the loop index
    int power = 1;      //"power" variable contains the i^th power of 3

    for(i = 1; ;i++){
        power = power * 3;
        if(power > a){
            return (i-1);
        }
    }
}
```

QC4

Part (a) [4 marks]

Express the return value f(n) as a mathematical function of n. Assume that n is a positive integer.

```
int f ( int n )
{
   int a, b, s;

   s = a = b = 0;
   while ((a < n) || (b < n)) {
      if (a <= b) ++a; else ++b;
      s += b;
   }
   return s;
}
```

Ans: n^2

Part (b) [6 marks]

Write a function find_power() which takes an integer a as argument and returns the smallest positive integer k for which $3^k \geqslant a$. Assume that $a \geqslant 3$. Do not use any math library function. You do not need to write any main() function.

Ans:

```
int find_power(int a) {
    int i;              //"i" is the loop index
    int power = 1;      //"power" variable contains the i^th power of 3
    for(i = 1; ;i++){
        power = power * 3;
        if(a <= power){
            return (i);
        }
    }
}
```

QD1

Part (a) [2 marks]

A function foo() is defined as follows.

```
int foo (int i) {
   int x = 1, j = 1;

   while (i) {
      x *= j;
      j++;
      if (j < 5) continue;
      else break;
   }
   return x;
}
```

What is returned if you call foo(6) from the main function?

Ans: 24

Part (b) [8 marks]

The following C program reads a sequence of positive integers until the user types -1. It counts the lengths of the non-decreasing subsequences (consecutively entered numbers), and prints the maximum length among them. For example, for input 6 7 2 29 17 5 5 11 16 6 7 8 -1, the non-decreasing subsequences are: {6, 7}, {2, 29}, {17}, {5, 5, 11, 16} and {6, 7, 8}. Thus the answer should be 4. Assume that the first integer read is not -1 and a single integer is a sequence of length 1. Complete the following code by filling in the blanks.

```
      int main ( )
      {
          int prevno, curno, curlength, maxlength;

          scanf ("%d", &prevno);
          scanf ("%d", &curno);

          curlength = maxlength = 1;

          while ( curno != -1 ) {
[A]          if ( __________ ) {
                 curlength++;
             } else {
[B]              if ( __________ ) {
                     maxlength = curlength;
                 }
[C]              curlength = __________ ;
             }
[D]          prevno = __________ ;
             scanf("%d", &curno );
          }

          if ( curlength > maxlength ) maxlength = curlength;

          printf ("Maximum length is %d\n", maxlength) ;
          return 0;
      }
```

Ans:

[A] `curno >= prevno`
[B] `curlength > maxlength`
[C] `1`
[D] `curno`

QD2

Part (a) [2 marks]

A function foo() is defined as follows.

```
int foo (int i) {
   int x = 1, j = 1;

   while (i) {
      x *= j;
      j++;
      if (j < 4) continue;
      else break;
   }
   return x;
}
```

What is returned if you call foo(6) from the main function?

Ans: 6

Part (b) [8 marks]

The following C program reads a sequence of positive integers until the user types -1. It counts the lengths of the non-increasing subsequences (consecutively entered numbers), and prints the maximum length among them. For example, for input 6 7 2 29 17 5 5 11 16 6 5 -1, the non-increasing subsequences are: {6}, {7, 2}, {29, 17, 5, 5}, {11}, {16, 6, 5}. Thus the answer should be 4. Assume that the first integer read is not -1 and a single integer is a sequence of length 1. Complete the following code by filling in the blanks.

```
      int main ( )
      {
          int prevno, curno, curlength, maxlength;

          scanf ("%d", &prevno);
          scanf ("%d", &curno);

          curlength = maxlength = 1;

          while ( curno != -1 ) {
[A]          if ( __________ ) {
                 curlength++;
             } else {
[B]              if ( __________ ) {
                     maxlength = curlength;
                 }
[C]              curlength = __________ ;
             }
[D]          prevno = __________ ;
             scanf("%d", &curno );
          }

          if ( curlength > maxlength ) maxlength = curlength;

          printf ("Maximum length is %d\n", maxlength) ;
          return 0;
      }
```

Ans:

[A] `curno <= prevno`
[B] `curlength > maxlength`
[C] `1`
[D] `curno`

QD3

Part (a) [2 marks]

A function foo() is defined as follows.

```
int foo (int i) {
   int x = 1, j = 1;

   while (i) {
      x *= j;
      j++; i--;
      if (j < 5) continue;
      else break;
   }
   return x;
}
```

What is returned if you call foo(6) from the main function?

Ans: 24

Part (b) [8 marks]

The following C program reads a sequence of positive integers until the user types -1. It counts the lengths of the strictly increasing subsequences (consecutively entered numbers), and prints the maximum length among them. For example, for input 6 7 2 29 17 5 5 11 16 6 7 8 -1, the strictly increasing subsequences are: {6, 7}, {2, 29}, {17}, {5}, {5, 11, 16} and {6, 7, 8}. Thus the answer should be 3. Assume that the first integer read is not -1 and a single integer is a sequence of length 1. Complete the following code by filling in the blanks.

```
      int main ( )
      {
          int prevno, curno, curlength, maxlength;

          scanf ("%d", &prevno);
          scanf ("%d", &curno);

          curlength = maxlength = 1;

          while ( curno != -1 ) {
[A]          if ( __________ ) {
                 curlength++;
             } else {
[B]              if ( __________ ) {
                     maxlength = curlength;
                 }
[C]              curlength = __________ ;
             }
[D]          prevno = __________ ;
             scanf("%d", &curno );
          }

          if ( curlength > maxlength ) maxlength = curlength;

          printf ("Maximum length is %d\n", maxlength) ;
          return 0;
      }
```

Ans:

[A] `curno > prevno`
[B] `curlength > maxlength`
[C] `1`
[D] `curno`

QD4

Part (a) [2 marks]

A function foo() is defined as follows.

```
int foo (int i) {
   int x = 1, j = 1;

   while (i) {
      x *= j;
      j++; i--;
      if (j < 4) continue;
      else break;
   }
   return x;
}
```

What is returned if you call foo(6) from the main function?

Ans: 6

Part (b) [8 marks]

The following C program reads a sequence of positive integers until the user types -1. It counts the lengths of the strictly decreasing subsequences (consecutively entered numbers), and prints the maximum length among them. For example, for input 6 7 2 29 17 5 5 11 16 6 5 -1, the strictly decreasing subsequences are: {6}, {7, 2}, {29, 17, 5}, {5}, {11}, {16, 6, 5}. Thus the answer should be 3. Assume that the first integer read is not -1 and a single integer is a sequence of length 1. Complete the following code by filling in the blanks.

```
      int main ( )
      {
          int prevno, curno, curlength, maxlength;

          scanf ("%d", &prevno);
          scanf ("%d", &curno);

          curlength = maxlength = 1;

          while ( curno != -1 ) {
[A]          if ( __________ ) {
                 curlength++;
             } else {
[B]              if ( __________ ) {
                     maxlength = curlength;
                 }
[C]              curlength = __________ ;
             }
[D]          prevno = __________ ;
             scanf("%d", &curno );
          }

          if ( curlength > maxlength ) maxlength = curlength;

          printf ("Maximum length is %d\n", maxlength) ;
          return 0;
      }
```

Ans:

[A] `curno < prevno`
[B] `curlength > maxlength`
[C] `1`
[D] `curno`

QE1

Part (a) [2 marks]

What value is printed by the following code?

```
int main()
{
   int a[8] = {3, 1, 0, 5, 2, 7, 6, 4};
   printf("%d", a[a[a[3]]]);
}
```

Ans: 4

Part (b) [8 marks]

Consider an array A containing n positive integers. We define an odd-left-rotate of A as a cyclic shift of ONLY the odd integers in A by one position to the left. The positions of the even integers in A remain unchanged. For example, if A = {2, 7, 3, 6, 4, 1, 8, 9}, then after the odd-left-rotate, A = {2, 3, 1, 6, 4, 9, 8, 7}. Fill in the blanks in the program below such that it does an odd-left-rotate of the array A. The variable firstpos will contain the index in A where the first odd integer is encountered, and lastnum will contain the last odd integer seen so far during the traversal.

```
      int main()
      {
          int firstpos, lastnum;
          int i, n, A[100], temp;

          scanf("%d", &n);
          for(i=0; i<n; i++) scanf("%d", &A[i]);

          firstpos = -1;

           /* Look at odd integers from the end */
[A]       for( __________ ) {
              if (A[i] % 2 == 1) {
[B]               if ( __________ ) {
                      firstpos = i;
[C]                   __________
                  } else {
                      temp = A[i];
                      A[i] = lastnum;
                      lastnum = temp;
                  }
              }
          }
           /* Do the last interchange */
[D]       __________
      }
```

Ans:

[A] `i=n-1; i>=0; i--`
[B] `firstpos == -1`
[C] `lastnum = A[i];`
[D] `A[firstpos] = lastnum;`

QE2

Part (a) [2 marks]

What value is printed by the following code?

```
int main()
{
   int a[8] = {3, 1, 0, 5, 7, 2, 6, 4};
   printf("%d", a[a[a[4]]]);
}
```

Ans: 7

Part (b) [8 marks]

Consider an array A containing n positive integers. We define an even-right-rotate of A as a cyclic shift of ONLY the even integers in A by one position to the right. The positions of the odd integers in A remain unchanged. For example, if A = {2, 7, 3, 6, 4, 1, 8, 9}, then after the even-right-rotate, A = {8, 7, 3, 2, 6, 1, 4, 9}. Fill in the blanks in the program below such that it does an even-right-rotate of the array A. The variable firstpos will contain the index in A where the first even integer is encountered, and lastnum will contain the last even integer seen so far during the traversal.

```
      int main()
      {
          int firstpos, lastnum;
          int i, n, A[100], temp;

          scanf("%d", &n);
          for(i=0; i<n; i++) scanf("%d", &A[i]);

          firstpos = -1;

          /* Look at even integers from the start */
[A]       for ( __________ ) {
              if (A[i] % 2 == 0) {
[B]               if ( __________ ) {
                      temp = A[i];
                      A[i] = lastnum;
                      lastnum = temp;
                  } else {
                      firstpos = i;
[C]                   __________
                  }
              }
          }
           /* Do the last interchange */
[D]       __________
      }
```

Ans:

[A] `i=0; i < n; i++`
[B] `firstpos != -1`
[C] `lastnum = A[i];`
[D] `A[firstpos] = lastnum;`

QE3

Part (a) [2 marks]

What value is printed by the following code?

```
int main()
{
   int a[8] = {2, 1, 5, 0, 3, 6, 7, 4};
   printf("%d", a[a[a[3]]]);
}
```

Ans: 5

Part (b) [8 marks]

Consider an array A containing n positive integers. We define an even-left-rotate of A as a cyclic shift of ONLY the even integers in A by one position to the left. The positions of the odd integers in A remain unchanged. For example, if A = {2, 7, 3, 6, 4, 1, 8, 9}, then after the even-left-rotate, A = {6, 7, 3, 4, 8, 1, 2, 9}. Fill in the blanks in the program below such that it does an even-left-rotate of the array A. The variable pos will contain the index in A where the first even integer is encountered, and num will contain the last even integer seen so far during the traversal.

```
      int main()
      {
          int pos, num;
          int i, k, n, A[100], temp;

          scanf("%d", &n);
          for(i=0; i<n; i++) scanf("%d", &A[i]);

          pos = -1; k = n-1;

           /* Look at even integers from the end */
[A]       while( __________ ) {
              if (A[k] % 2 == 0) {
[B]               if ( __________ ) {
                      pos = k;
[C]                   __________
                  } else {
                      temp = A[k];
                      A[k] = num;
                      num = temp;
                  }
              }
              k--;
          }
           /* Do the last interchange */
[D]       __________
      }
```

Ans:

[A] `k >= 0`
[B] `pos == -1`
[C] `num = A[k];`
[D] `A[pos] = num;`

Session 2

25-May-2021

10:30am – 11:45pm

Sections 2, 7, 8, 9, 10

QF1

Part (a) [2 marks]

Consider the following declarations:

```
float x = 12.0, y = 6.0;
int a = 4, b = 3;
float z;
```

What is the value of z computed by the following assignment statement?

```
z = (a / b) > 1 ? x : y;
```

Ans: 6.000000

Part (b) [8 marks]

Let S and T be two null-terminated strings consisting of lower-case letters only. T is called a substring of S if all the symbols in T appear in the same order contiguously in S. For example, "abca" is a substring of "bbabcabcabac". Your task is to find whether T appears in S as a substring, and if yes, write in R a null-terminated string that is obtained from S by replacing the leftmost occurrence of T in S by converting the letters of T to upper case. For the above example, R should store the string "bbABCAbcabac". If T is not present in S as a substring, then R should be the same as S. You are not permitted to use any other variables.

```
[A]   void ucfirst ( __________ )  /* Pass S, T and R */
      {
         int n, m, i, j, k;

         /* Compute the lengths of S and T */
         n = strlen(S); m = strlen(T);

         strcpy(R,S); /* Initial copy */

[B]      for ( __________ ) {  /* Loop on i */
            /* Try to match T with S[i...i+m-1] */
            for (j=0; j<m; ++j) {
               if ( S[i+j] != T[j]) break;
            }

[C]         if ( __________ ) { /* if match found */
               /* Change only the matching part in R */
[D]            for (k=i; k<i+m; ++k) R[k] = __________ ;
               return;
            }
         }
      }
```

Ans:

[A] `char S[], char T[], char R[]`
[B] `i=0; i<=n-m; ++i`
[C] `j == m`
[D] `'A' + (R[k] - 'a')`

QF2

Part (a) [2 marks]

Consider the following declarations:

```
float x = 6.0, y = 12.0;
int a = 7, b = 3;
float z;
```

What is the value of z computed by the following assignment statement?

```
z = (a / b) != 2 ? x : y;
```

Ans: 12.000000

Part (b) [8 marks]

Let S and T be two null-terminated strings consisting of lower-case letters only. T is called a substring of S if all the symbols in T appear in the same order contiguously in S. For example, "abca" is a substring of "bbabcabcabac". Your task is to find whether T appears in S as a substring, and if yes, write in R a null-terminated string that is obtained from S by replacing the rightmost occurrence of T in S by converting the letters of T to upper case. For the above example, R should store the string "bbabcABCAbac". If T is not present in S as a substring, then R should be the same as S. You are not permitted to use any other variables.

```
[A]   void uclast ( __________ )  /* Pass S, T and R */
      {
         int n, m, i, j, k;

         /* Compute the lengths of S and T */
         n = strlen(S); m = strlen(T);

         strcpy(R,S); /* Initial copy */

[B]      for ( __________ ) {  /* Loop on i */
            /* Try to match T with S[i...i+m-1] */
            for (j=0; j<m; ++j) {
               if ( S[i+j] != T[j]) break;
            }

[C]         if ( __________ ) { /* if match found */
               /* Change only the matching part in R */
[D]            for (k=i; k<i+m; ++k) R[k] = __________ ;
               return;
            }
         }
      }
```

Ans:

[A] `char S[], char T[], char R[]`
[B] `i=n-m; i>=0; --i`
[C] `j == m`
[D] `'A' + (R[k] - 'a')`

QF3

Part (a) [2 marks]

Consider the following declarations:

```
float x = 12.0;
int a = 4, b = 3;
float z;
```

What is the value of z computed by the following assignment statement?

```
z = (int) (x + b) / a % b ;
```

Ans: 0.000000

Part (b) [8 marks]

Let S and T be two null-terminated strings consisting of lower-case letters only. T is called a substring of S if all the symbols in T appear in the same order contiguously in S. For example, "abca" is a substring of "bbabcabcabac". Your task is to find whether T appears in S as a substring, and if yes, write in R a null-terminated string that is obtained from S by replacing the leftmost occurrence of T in S by dots. For the above example, R should store the string "bb....bcabac". If T is not present in S as a substring, then R should be the same as S. You are not permitted to use any other variables.

```
[A]   void rpfirst ( __________ )  /* Pass S, T and R */
      {
         int n, m, i, j, k;

         /* Compute the lengths of S and T */
         n = strlen(S); m = strlen(T);

         strcpy(R,S); /* Initial copy */

[B]      for ( __________ ) {  /* Loop on i */
            /* Try to match T with S[i...i+m-1] */
            for (j=0; j<m; ++j) {
               if ( S[i+j] != T[j]) break;
            }

[C]         if ( __________ ) { /* if match found */
               /* Change only the matching part in R */
[D]            for (k=i; k<i+m; ++k) R[k] = __________ ;
               return;
            }
         }
      }
```

Ans:

[A] `char S[], char T[], char R[]`
[B] `i=0; i<=n-m; ++i`
[C] `j == m`
[D] `'.'`

QF4

Part (a) [2 marks]

Consider the following declarations:

```
float x = 12.0;
int a = 4, b = 5;
float z;
```

What is the value of z computed by the following assignment statement?

```
z = (int) (x + b) / b % a ;
```

```
--------------------
Ans: 3.000000
--------------------
```

Part (b) [8 marks]

Let S and T be two null-terminated strings consisting of lower-case letters only. T is called a substring of S if all the symbols in T appear in the same order contiguously in S. For example, "abca" is a substring of "bbabcabcabac". Your task is to find whether T appears in S as a substring), and if yes, write in R a null-terminated string that is obtained from S by replacing the rightmost occurrence of T in S by dots. For the above example, R should store the string "bbabc....bac". If T is not present in S as a substring, then R should be the same as S. You are not permitted to use any other variables.

```
[A]   void rplast ( __________ )  /* Pass S, T and R */
      {
         int n, m, i, j, k;

         /* Compute the lengths of S and T */
         n = strlen(S); m = strlen(T);

         strcpy(R,S); /* Initial copy */

[B]      for ( __________ ) {  /* Loop on i */
            /* Try to match T with S[i...i+m-1] */
            for (j=0; j<m; ++j) {
               if ( S[i+j] != T[j]) break;
            }

[C]         if ( __________ ) { /* if match found */
               /* Change only the matching part in R */
[D]            for (k=i; k<i+m; ++k) R[k] = __________ ;
               return;
            }
         }
      }
```

Ans:

[A] `char S[], char T[], char R[]`
[B] `i=n-m; i>=0; --i`
[C] `j == m`
[D] `'.'`

QG1

Part (a) [2 marks]

What will be printed by the following program fragment? Assume that A is a character array containing the null-terminated string "378" and i and j are int type variables initialized to 0.

```
while (j < strlen(A)) {
   i = 10*i + A[j++] - '0';
}
printf("%d", i);
```

Ans: 378

Part (b) [8 marks]

Fill in the blanks to complete the following program that takes a string S as input (assume that the string can be at most 100 characters long and can contain any character), and creates another string R which is an encoded version of S. R is said to be an encoding of S if every digit k in S is replaced by two letters in lower case, the k-th letter after 'a' and the k-th letter before 'z'. All other characters are ignored. For example, if S = "031a2#", then R = "azdwbycx". You cannot use any string functions or use any ascii values directly.

```
int main()
{
    char S[101], R[201];
    int i, j;

    scanf("%s", S);

    i = j = 0;

     /* Loop over S */
[A] while ( __________ ) {
[B]     if ( (S[j] < '0') || (S[j] > '9') ) { j++; __________ }
         /* Encode in R */
[C]     R[i] = __________ ;
[D]     R[i+1] = __________ ;
         j = j+1;
[E]     i = __________ ;
    }

    R[i] = '\0';

    printf("R = %s\n", R);
}
```

Ans:

[A] `S[j] != '\0'`
[B] `continue;`
[C] `'a' + S[j] - '0'`
[D] `'z' - (S[j] - '0')`
[E] `i+2`

QG2

Part (a) [2 marks]

What will be printed by the following program fragment? Assume that A is a character array containing the null-terminated string "dbh" and i and j are int type variables initialized to 0.

```
while (j < strlen(A)) {
   i = 10*i + A[j++] - 'a';
}
printf("%d", i);
```

Ans: 317

Part (b) [8 marks]

Fill in the blanks to complete the following program that first takes two strings S and R as input. Assume that each of R and S has at most 100 characters, all characters in S are distinct, and all characters in R are distinct. The program then deletes any character in S that also occurs in R, and prints the modified S. For example, if S = "Dahu#@rn{cKP)e" and R = "gsNer{hP}wpE~@", then S should be modified to "Dau#ncK)". You cannot use any other arrays.

```
int main()
{
    char R[101], S[101];
    int i, j, k;

    scanf("%s%s", S, R);

     /* Check if each character in R is in S or not */
[A] for ( _________ ) {
        for (j=0; S[j] != '\0'; j++) {
[B]         if ( _________ ) break;
        }
[C]     if ( _________ ) {
            /* Delete R[i] from S. Since chars are distinct, only one occurrenceof R[i] in S */
[D]         for ( _________ )  S[k-1] = S[k];
[E]         S[k-1] = _________ ;
        }
    }

    printf("Modified S = %s\n", S);
}
```

Ans:

[A] `i=0; R[i] != '\0'; i++`
[B] `R[i] == S[j]`
[C] `S[j] != '\0'`
[D] `k=j+1; S[k] != '\0'; k++`
[E] `'\0'`

QG3

Part (a) [2 marks]

What will be printed by the following program fragment? Assume that A is a character array containing the null-terminated string "dbh" and i and j are int type variables initialized to 0 and strlen(A)-1, respectively.

```
while (j >= 0) {
   i = 10*i + A[j--] - 'a';
}
printf("%d", i);
```

Ans: 713

Part (b) [8 marks]

Fill in the blanks to complete the following program that reads a string S as input (assume that S has at most 50 characters), and removes everything other than alphabetic letters from it. It finally prints the modified S. For example, if S = "?weF@234i!HDsK_0c/" initially, then it should be modified to "weFiHDsKc". You cannot use any other arrays.

```
      int main()
      {
          char S[51];
          int i, j;

          scanf("%s", S);

         /*  Loop over S to find non-alphabet characters */
          i = 0;
[A]       while ( __________ )     {
              if (S[i] < 'A' || S[i] > 'Z') {
[B]               if ( __________ ) {
                       /* Non-alphabet character. Remove from S */
[C]                for ( __________ ) S[j-1] = S[j];
[D]                S[j-1] = __________ ;
[E]                i = __________ ;
                  }
              }
              i++;
          }

          printf("Modified S = %s\n", S);
      }
```

Ans:

[A] `S[i] != '\0'`
[B] `S[i] < 'a' || S[i] > 'z'`
[C] `j=i+1; S[j] != '\0'; j++`
[D] `'\0'`
[E] `i - 1`

QH1

Part (a) [4 marks]

Consider the following function. Assume that n is a positive integer.

```
int f ( int n, int a, int b )
{
   while (n >= a) n -= a;
   if (n >= b) n -= b;
   return n;
}
```

For which values of n, is 2 returned by f(n,6,2)?

Ans: All integers n of the form 6k + 4

Part (b) [6 marks]

Write a function print_a() that takes an integer n as an argument and prints all integers a such that $1 \leqslant a < n$ and $2a^2 + 1$ is prime. Do not use any math library function. If you want, you may write a function for checking prime numbers, but do not write any main function.

Eg. For n = 10, Output: 1, 3, 6, 9

Ans:

```
int is_prime(int n) {
    int i, flag = 0;

    for (i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(n == 1)
        return -1;
    else if (flag == 0)
        return 1;
    else
        return 0;
}

void print_a(int n){
    int i, res;

    for(i = 1; i<n; i++){
        res = is_prime(2*i*i + 1);
        if(res == 1)
            printf("%d ", i);
    }
}
```

QH2

Part (a) [4 marks]

Consider the following function. Assume that n is a positive integer.

```
int f ( int n, int a, int b )
{
   while (n >= a) n -= a;
   if (n >= b) n -= b;
   return n;
}
```

For which values of n, is 2 returned by f(n,4,3)?

Ans: All integers n of the form 4k + 2

Part (b) [6 marks]

Write a function print_a() that takes an integer n as an argument and prints all integers a such that $1 \leqslant a < n$ and $2a^2 - 1$ is prime. Do not use any math library function. If you want, you may write a function for checking prime numbers, but do not write any main function.

Eg. For n = 10, Output: 2, 3, 4, 6, 7, 8

Ans:

```
int is_prime(int n) {
    int i, flag = 0;

    for (i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(n == 1)
        return -1;
    else if (flag == 0)
        return 1;
    else
        return 0;
}

void print_a(int n){
    int i, res;

    for(i = 1; i<n; i++){
        res = is_prime(2*i*i - 1);
        if(res == 1)
            printf("%d ", i);
    }
}
```

QH3

Part (a) [4 marks]

Consider the following function. Assume that n is a positive integer.

```
int f ( int n, int a, int b )
{
   while (n >= a) n -= a;
   if (n >= b) n -= b;
   return n;
}
```

For which values of n, is 2 returned by f(n,5,2)?

Ans: All integers n of the form 5k + 4

Part (b) [6 marks]

Write a function print_a() that takes an integer n as an argument and prints all integers a such that $1 \leqslant a < n$ and $3a^2 - 1$ is prime. Do not use any math library function. If you want, you may write a function for checking prime numbers, but do not write any main function.

Eg. For n = 10, Output: 1 2 4 6 8

Ans:

```
int is_prime(int n) {
    int i, flag = 0;

    for (i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(n == 1)
        return -1;
    else if (flag == 0)
        return 1;
    else
        return 0;
}

void print_a(int n){
    int i, res;

    for(i = 1; i<n; i++){
        res = is_prime(3*i*i - 1);
        if(res == 1)
            printf("%d ", i);
    }
}
```

QH4

Part (a) [4 marks]

Consider the following function. Assume that n is a positive integer.

```
int f ( int n, int a, int b )
{
   while (n >= a) n -= a;
   if (n >= b) n -= b;
   return n;
}
```

For which values of n, is 2 returned by f(n,5,3)?

Ans: All integers n of the form 5k + 2

Part (b) [6 marks]

Write a function print_a() that takes an integer n as an argument and prints all integers a such that $1 \leqslant a < n$ and $3a^2 + 1$ is prime. Do not use any math library function. If you want, you may write a function for checking prime numbers, but do not write any main function.

Eg. For n = 10, Output: 2, 6, 8

Ans:

```
int is_prime(int n) {
    int i, flag = 0;

    for (i = 2; i <= n / 2; i++) {
        if (n % i == 0) {
            flag = 1;
            break;
        }
    }

    if(n == 1)
        return -1;
    else if (flag == 0)
        return 1;
    else
        return 0;
}

void print_a(int n){
    int i, res;

    for(i = 1; i<n; i++){
        res = is_prime(3*i*i + 1);
        if(res == 1)
            printf("%d ", i);
    }
}
```

Q11

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int i, j = 10, y = 0;
   for (i = 0; i < j; i++)
      while(j) y += j--;
   printf("%d", y);
}
```

Ans: 55

Part (b) [8 marks]

The program given below is meant for printing a triangular pattern. The program reads the number of rows from the user, and prints the pattern using stars (*) and spaces. The pattern for rows = 8 is given below.

```
       *
      * *
     * * *
    * * * *
   * * * * *
  * * * * * *
 * * * * * * *
* * * * * * * *
```

Fill in the blanks to complete the code.

```
      int main() {
         int i, space, rows, k;

         printf("Enter the number of rows: ");
         scanf("%d", &rows);

         /* Loop on i for printing individual rows */
[A]      for ( __________ ) {
            /* First print the initial spaces (loop on variable space) */
[B]         for ( __________ )  printf(" ");

            /* Then print the stars (loop on variable k) */
            k = 0;
[C]         while ( __________ ) {
[D]            printf( __________ ) ;
[E]            __________
            }

            printf("\n");
         }

         return 0;
      }
```

Ans:

[A] `i = 1; i <= rows; ++i`
[B] `space = 1; space <= rows - i; ++space`
[C] `k < i`
[D] `"* "`
[E] `++k;`

QI2

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int i, j = 10, y = 0;
   for (i = 0; i < j; i++)
      while(j) y += j--;
   printf("%d", y);
}
```

Ans: 55

Part (b) [8 marks]

The program given below is meant for printing a triangular pattern. The program reads the number of rows from the user, and prints the pattern using stars (*) and spaces. The pattern for rows = 8 is given below.

```
* * * * * * * *
 * * * * * * *
  * * * * * *
   * * * * *
    * * * *
     * * *
      * *
       *
```

Fill in the blanks to complete the code.

```
      int main() {
         int i, space, rows, k;

         printf("Enter the number of rows: ");
         scanf("%d", &rows);

         /* Loop on i for printing individual rows */
[A]      for ( __________ ) {
            /* First print the initial spaces (loop on variable space) */
[B]         for ( __________ )  printf(" ");

            /* Then print the stars (loop on variable k) */
            k = 0;
[C]         while ( __________ ) {
[D]            printf( __________ ) ;
[E]            __________
            }

            printf("\n");
         }

         return 0;
      }
```

Ans:

[A] `i = rows; i >= 1; --i`
[B] `space = 1; space <= rows - i; ++space`
[C] `k < i`
[D] `"* "`
[E] `++k;`

QI3

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int i, j = 10, y = 0;
   for (i = 0; i < j; i++)
      while (j) y += --j;
   printf("%d", y);
}
```

Ans: 45

Part (b) [8 marks]

The program given below is meant for printing a triangular pattern. The program reads the number of rows from the user, and prints the pattern using stars (*) and spaces. The pattern for rows = 8 is given below.

```
       *
      * *
     * * *
    * * * *
   * * * * *
  * * * * * *
 * * * * * * *
* * * * * * * *
```

Fill in the blanks to complete the code.

```
      int main() {
         int i, space, rows, k;

         printf("Enter the number of rows: ");
         scanf("%d", &rows);

         /* Loop on i for printing individual rows */
[A]      for ( __________ ) {
            /* First print the initial spaces (loop on variable space) */
[B]         for ( __________ )  printf(" ");

            /* Then print the stars (loop on variable k) */
            k = 0;
[C]         while ( __________ ) {
[D]            printf( __________ ) ;
[E]            __________
            }

            printf("\n");
         }

         return 0;
      }
```

Ans:

[A] `i = 1; i <= rows; ++i`
[B] `space = 1; space <= rows - i; ++space`
[C] `k < i`
[D] `"* "`
[E] `++k;`

QI4

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int i, j = 10, y = 0;
   for (i = 0; i < j; i++)
      while (j) y += --j;
   printf("%d", y);
}
```

Ans: 45

Part (b) [8 marks]

The program given below is meant for printing a triangular pattern. The program reads the number of rows from the user, and prints the pattern using stars (*) and spaces. The pattern for rows = 8 is given below.

```
* * * * * * * *
 * * * * * * *
  * * * * * *
   * * * * *
    * * * *
     * * *
      * *
       *
```

Fill in the blanks to complete the code.

```
      int main() {
         int i, space, rows, k;

         printf("Enter the number of rows: ");
         scanf("%d", &rows);

         /* Loop on i for printing individual rows */
[A]      for ( __________ ) {
            /* First print the initial spaces (loop on variable space) */
[B]         for ( __________ )  printf(" ");

            /* Then print the stars (loop on variable k) */
            k = 0;
[C]         while ( __________ ) {
[D]            printf( __________ ) ;
[E]            __________
            }

            printf("\n");
         }

         return 0;
      }
```

Ans:

[A] `i = rows; i >= 1; --i`
[B] `space = 1; space <= rows - i; ++space`
[C] `k < i`
[D] `"* "`
[E] `++k;`

QJ1

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int a[8] = {3, 1, 0, 5, 2, 7, 6, 4};
   printf("%d", a[a[3]+2] - a[3] - a[2]);
}
```

Ans: -1

Part (b) [8 marks]

Consider an array A containing an even number of positive integers, exactly half of which are odd and half are even. We define A to be balanced if every odd index contains an odd integer and every even index contains an even integer. For example, the array A = {2, 7, 4, 1, 6, 7} is balanced (recall that array indices start from 0), while the array A = {2, 7, 1, 3, 6, 8} is not since, for example, A[2] contains an odd integer. Fill in the blanks in the program below such that it converts the array A into a balanced array (assume that n will be entered as an even positive integer and the integers entered in A will have n/2 even and n/2 odd integers).

```
      int main()
      {
          int i, j, n, A[100], temp;

          scanf("%d", &n);
          for (i=0; i<n; i++) scanf("%d", &A[i]);

          i = 1; j = 0;
          while (1) {
              /* Scan for the first out-of-place odd integer and first out-of-place even integer
                 using i and j in two loops */
[A]           while ( __________ ) i = i + 2;

[B]           while ( __________ ) j = j + 2;

              /* End the while(1) loop if all odd indices have odd integers and all even indices
                 have even integers, else do the needful */
[C]           if ( __________ ) break;


              temp = A[i];
[D]           __________
              A[j] = temp;
          }
      }
```

Ans:

[A] `(i < n) && (A[i] % 2 == 1)`
[B] `(j < n) && (A[j] % 2 == 0)`
[C] `i >= n || j >= n`
[D] `A[i] = A[j];`

QJ2

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int a[8] = {3, 1, 0, 5, 2, 7, 6, 4};
   printf("%d", a[a[4]+3] - a[3] - a[4]);
}
```

Ans: 0

Part (b) [8 marks]

Consider an array A containing an even number of positive integers, exactly half of which are odd and half are even. We define A to be anti-balanced if every odd index contains an even integer and every even index contains an odd integer. For example, the array A = {1, 4, 7, 2, 5, 6} is anti-balanced (recall that array indices start from 0), while the array A = {7, 6, 8, 2, 3, 1} is not since, for example, A[2] contains an even integer. Fill in the blanks in the program below such that it converts the array A into an anti-balanced array (assume that n will be entered as an even positive integer and the integers entered in A will have n/2 even and n/2 odd integers).

```
      int main()
      {
          int i, j, n, A[100], temp;

          scanf("%d", &n);
          for (i=0; i<n; i++) scanf("%d", &A[i]);

          i = 1; j = 0;
          while (1) {
               /* Scan for the first out-of-place odd integer and first out-of-place even integer
                  using i and j in two loops */
[A]           while ( __________ ) i = i + 2;

[B]           while ( __________ ) j = j + 2;

              /* End the while(1) loop if all odd indices have even integers and all even indices
                 have odd integers, else do the needful */
[C]           if ( __________ ) {
                  temp = A[j];
[D]               _________
                  A[i] = temp;
              } else
                  break;
          }
      }
```

Ans:

[A] `(i < n) && (A[i] % 2 == 0)`
[B] `(j < n) && (A[j] % 2 == 1)`
[C] `i < n && j < n`
[D] `A[j] = A[i];`

QJ3

Part (a) [2 marks]

What is printed by the following program?

```
int main()
{
   int a[8] = {3, 1, 0, 5, 2, 7, 6, 4};
   printf("%d", a[a[3]-2] + a[3] + a[2]);
}
```

Ans: 10

Part (b) [8 marks]

Consider an array A containing an even number of integers, exactly half of which are positive and half are negative. We define A to be sign-balanced if every odd index contains a positive integer and every even index contains a negative integer. For example, the array A = {-2, 5, -8, 1, -6, 5} is sign-balanced (recall that array indices start from 0), while the array A = {-2, 5, 1, 10, -6, -5} is not since, for example, A[2] contains a positive integer. Fill in the blanks in the program below such that it converts the array A into a sign-balanced array (assume that n will be entered as an even positive integer and the integers entered in A will have n/2 positive integers and n/2 negative integers).

```
      int main()
      {
          int i, j, n, A[100], temp;

          scanf("%d", &n);
          for(i=0; i<n; i++) scanf("%d", &A[i]);

          i = 1; j = 0;
          while (1) {
               /* Scan for the first out-of-place positive integer and the first out-of-place negative
                  integer using i and j in two loops */
[A]           while ( __________ ) i = i + 2;

[B]           while ( __________ ) j = j + 2;

              /* End the while(1) loop if all odd indices have positive integers and all even indices
                 have negative integers */
[C]           if ( __________ ) break;

              temp = A[i];
[D]           _________
              A[j] = temp;
          }
      }
```

Ans:

[A] `(i < n) && (A[i] > 0)`
[B] `(j < n) && (A[j] < 0)`
[C] `i >= n || j >= n`
[D] `A[i] = A[j];`
