Indian Institute of Technology Kharagpur
Programming and Data Structures (CS10001)
Autumn 2017-18: End-Semester Examination

Time: 3 Hours **Full Marks: 100**

INSTRUCTIONS

1. Answer ALL questions.
2. Please write the answers either within the boxes provided or on the blank lines to be filled up. Any answer written elsewhere will not be evaluated.
3. You may use the blank pages at the end for your rough work.
4. In the C programs, **assume that all appropriate header files are included.**

QUESTION 1. **Answer the following questions as directed.**

a) Given that the ASCII codes for lowercase letters in the English alphabet ‘a’ to ‘z’ are 97 to 122 (in decimal), what will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int a;
  a = ‘d’ * 2 + 4;
  printf(“%d”, a);
  return 0;
}
```

204

b) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int a = 5;
  printf(“%d”, a == 5 ? 0 : 5);
  return 0;
}
```

0

c) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int a = 0;
  printf(“%d”, a == 0);
  return 0;
}
```

1

d) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int a, sum = 0;
  for (a = 0; a < 5; a++)
    sum = sum + a++;
  printf(“%d”, sum);
  return 0;
}
```

6

e) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int a[4] = {10, 20, 30, 40}, count, sum = 0;
  for (count = 0; count < 4; count++)
    sum = sum + a[count] + a[3-count];
  printf("%d", sum);
  return 0;
}
```

```
200
```

f) What will be displayed when the following C program is executed? [1 Mark]

```
int a = 1;
void myProc() {
  if ( a != 2 ) {
    int a = 3; printf ("%d\n", a);
  }
  else { int a = 4; }
  printf ( "%d\n", a);
}
int main() {
  myProc(); printf ("%d\n", a);
  return 0;
}
```

```
3
1
1
```

g) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int *a, **b, c;
  b = &a; a = &c; c = 20;
  printf ("%d %d", **b, *a);
  return 0;
}
```

```
20 20
```

h) What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  int *a, b[5] = {1, 2, 3, 4, 5};
  a = b;
  printf ("%d %d", *(b+2), *(a+3));
  return 0;
}
```

```
3 4
```

i) Write a **single** C statement that declares **a** to be an integer type variable, **b** to be an integer pointer, and **b** is initialized to contain the address of **a**. [1 Mark]

```
int a, *b = &a;
```

j) Consider that a text file named **file1.txt** contains a single line with just one word **Avishek** in it. What will be displayed when the following C program is executed? [1 Mark]

```
int main() {
  FILE *fptr; char line[6];
  if ((fptr = fopen("file1.txt", "r")) != NULL)
    while (fgets(line, 3, fptr) != NULL)
      printf ("%s\n",line);
  return 0;
}
```

```
Av
is
he
k
```

QUESTION 2. **Answer the following questions as directed.**

a) Write a **single** C statement that declares **str1** to be an array of three pointers pointing to constant strings "Sarthak", "Anita" and "Dinesh", respectively. [2 Marks]

```
char *str1[ ] = {"Sarthak", "Anita", "Dinesh"}; OR
char *str1[3] = {"Sarthak", "Anita", "Dinesh"};
```

b) What will be displayed when the following C program is executed? [2 Marks]

```
int main() {
  int i;
  for (i = 0; i < 4; i++) {
    int x = 0;
    if (x==2) break;
    else printf ("%d\n",x);
    x++;
  }
  return 0;
}
```

```
0
0
0
0
```

c) What will be displayed when the following C program is executed? [2 Marks]

```
struct abc {int a; int b;};
struct abc def[5];
int main() {
  def[0].a = 5; def[0].b = 10;
  def[1].a = 15; def[1].b = 20;
  printf ("%d\n", ++(*def).a);
  return 0;
}
```

```
6
```

d) What will be displayed when the following C program is executed? [2 Marks]

```
int main() {
  char *abc = "bef";
  if (*abc + 1 == 'c') printf ("%c", *abc++);
  else printf ("%s", abc);
  return 0;
}
```

```
b
```

e) What is the hexadecimal representation of the decimal number 140.5? [2 Marks]

```
8C.8
```

f) What is the decimal representation of the result if 11101100 is subtracted from 01010000? Assume that both the binary numbers are represented in 8-bit 2's complement form. [2 Marks]

```
100
```

g) What will be the displayed when the following C program is executed? [2 Marks]

```
int main() {
  char b = 'a';
  switch (b) {
    default: printf ("10\n");
    case 'a':
    case 'A': printf ("20\n");
    case 'b':
    case 'B': printf ("30\n");
              break;
  }
  return 0;
}
```

```
20
30
```

h) What will be displayed when the following C program is executed if each address is of size 8 bytes in the computer? [2 Marks]

```
int main() {
  char a[5] = {'d', 'e', 'f', 'g', 'h'};
  printf ("%u %u\n", (a + 3) - a, *(a + 3) - *a);
  return 0;
}
```

```
3 3
```

i) What will be displayed when the following C program is executed from the command line as:
./a.out A 5 [2 Marks]

```
int main(int argc, char *argv[]) {
  char c; int a;
  sscanf(argv[1], "%c", &c);
  sscanf(argv[2], "%d", &a);
  printf ("%c", c + a);
  return 0;
}
```

```
F
```

j) What will be displayed when the following C program is executed? [2 Marks]

```
struct node {
   int data;  struct node *next;
};
int main() {
  struct node a, b, c;
  a.data = 10; b.data = 20; c.data = 30;
  a.next = &b; b.next = &c; c.next = NULL;
  printf ("%d %d", a.data, a.next→next→data);
  return 0;
}
```

```
10 30
```

Answer the following questions as directed.

QUESTION 3.

a) The following C function is used to compute the product of two matrices, with each matrix represented as a 2-d array. The function takes as parameters a 2-d array A with n_r_A rows and n_c_A columns and a 2-d array B with n_r_B rows and n_c_B columns. The function returns a pointer to the first element of the product matrix C. Fill in the missing lines. [1+1+1+1+1+1 = 6 Marks]

```
int **mult(int **a, int n_r_a, int n_c_a, int **b, int n_r_b, int n_c_b) {

  int **c     ;                //Declare suitable variable for returning result
  int sum, i, j, k;

  c = (int **) malloc (sizeof(int *)*n_r_a);
  for (i = 0; i < n_r_a; i++) {

    c[i] = (int *) malloc(sizeof(int)*n_c_b);
    for (j = 0; j < n_c_b; j++) {
      c[i][j] = 0;
      for (k = 0; k < n_c_a           ; k++)

        c[i][j] = c[i][j] + a[i][k] * b[k][j];
      }
  }
  return c;
}
```

b) The following C program is written to compute the sum of all the elements of a 2D array. However, when compiled, a **syntax error** is detected. Encircle the line where there is a syntax error and write it correctly in the box provided. Even after correcting the syntax error, when the program is executed, the sum is not printed correctly since there is also a **logical error** in one of the statements in the **get_sum** function. Encircle the statement which has a logical error and write it correctly in the box provided. [2+2=4 Marks]

```
int get_sum (int x[2][])  {
  int sum = 0, i, j;
  for (i = 0; i < 2; i++)
    for (j = 0; j < 3; j++)
      sum =+ x[i][j];
  return sum;
}
int main() {
  int A[2][3] = {{1, 2, 3}, {4, 5, 6}};
  printf ("%d\n", get_sum(A));
  return 0;
}
```

```
int get_sum(int x[ ][3]) {

sum += x[i][j];
```

QUESTION 4.

a) The following program uses quicksort to sort an array of integers in decreasing order. What will be displayed when the program is executed? [5 Marks]

```
int partition (int *A, int lt, int rt) {
  int pivot, i, j, temp;
  pivot = A[lt]; i = lt; j = rt;
  while (i < j) {
    while (A[j] < pivot) j--;  while (A[i] >= pivot && i <= rt) i++;
    if (i < j) {
      temp = A[j]; A[j] = A[i]; A[i] = temp; j--;
    }
  }
  temp = A[lt]; A[lt] = A[j]; A[j] = temp;
  return j;
}
void quicksort (int *A, int lt, int rt) {
  int ind;
  if (lt >= rt) return;
  ind = partition(A,lt,rt); printf ("%d\n", A[ind]);
  quicksort(A,ind+1,rt); quicksort(A,lt,ind-1);
}
int main() {
  int X[9] = {45, 65, 24, 38, 17, 101, 4, 203, 115};
  quicksort (X, 0, 8);
  return 0;
}
```

```
45
17
38
101
115
```

b) The following **recursive** function is used to implement insertion sort. Fill in the missing lines.

[1+1+1+1+1=5 Marks]

```
void recur_insertion_sort (int A[], int n) {
  int j, key;

  if (n <= 1) return;                          //Base case

       //Recursively sort first n-1 elements

  recur_insertion_sort (A, n-1);
       // Insert the last element at its correct position
  key = A[n-1];
       // Move each array element greater than key one position to its right

  j = n-2;
  while (j >= 0 && A[j] > key) {

    A[j+1] = A[j];
    j--;
  }
  A[j+1] = key;
}
```

QUESTION 5.

a) When the following C program is executed, it produces seven lines of output. The first, third, fourth and fifth lines of the output are shown in the box below (the addresses are printed in hex). Write what will be printed as the second, sixth and seventh lines of the output. Assume that each address is of size 4 bytes in the computer.

[2+2+2=6 Marks]

```
int **allocate (int h, int w) {
  int **p, i, j;
  p = (int **) malloc(h*sizeof (int *));
  printf ("p = %p\n", p);
  printf ("p+1 = %p\n", p+1);
  for (i = 0; i < h; i++) p[i] = (int *) malloc(w * sizeof(int));
  for (i = 0; i < h; i++) printf("p[%d] = %p\n", i, p[i]);
  return(p);
}

int main() {
  int **q;
  int M = 3, N = 3;
  q = allocate (M, N);
  printf ("&q[1][2] = %p\n", &q[1][2]);
  printf ("&q[0] = %p\n", &q[0]);
  return 0;
}
```

```
p = 00980FF8

p+1 = 00980FFC
p[0] = 00981010
p[1] = 00981028
p[2] = 00981040

&q[1][2] = 00981030

&q[0] = 00980FF8
```

b) What will be displayed when the following C program is executed? [2+2=4 Marks]

```
struct abc {int a; int b;} *def;
int main() {
  char *ch = (char *) malloc (4*sizeof(int));
  def = (struct abc *) malloc(5*sizeof(struct abc));
  def→a = 5; def→b = 10;
  def[1].a = 10; def[1].b = 11;
  ++def→a;
  ++def;
  printf ("%d\n", def→a);
  printf ("%d\n", sizeof(def) - sizeof(ch));
  return 0;
}
```

```
10
0
```

QUESTION 6.

a) What will be displayed when the following C program is executed? [1+1+2+2=6 Marks]

```
int rec1 (int);
int rec2 (int);
int main() {
  printf("%d : %d : %d : %d\n", rec1(0), rec1(1), rec2(1022), rec2(2001));
  return 0;
}
int rec1 (int n) {
  if (n == 0) return 1;
  else return ( rec2(n-1) );
}
int rec2 (int m) {
  return ( !rec1(m) );
}
```

```
1 : 0 : 0 : 1
```

b) The following C function **recursively** computes the product of two input parameters **a** and **b**. Fill in the missing lines. [2+2=4 Marks]

```
int recur (int a, int b) {
  if (b == 0) return 0;

  if (b % 2 == 0) return recur (a+a, b/2);

  return recur (a+a, b/2) + a;
}
```

QUESTION 7.

a) We want to convert the cartesian form (**a**, **b**) of a complex number **a** + i**b** to the corresponding polar form **r**(cos**θ** + i sin**θ**) representation denoted as (**r**, **θ**). Here **r** and **θ** are related to **a** and **b** as: **r** = $(\mathbf{a}^2 + \mathbf{b}^2)^{1/2}$ and **θ** = $\tan^{-1}$(**b**/**a**) in radian. For example, if the complex number 3 + i4 is represented in cartesian form as (3, 4), its polar form representation is (5, 0.927295) since $(3^2+4^2)^{1/2} = 5$ and $\tan^{-1}(4/3) = 0.927295$ radian.

The following C program uses a function **cart2pol** to convert from cartesian to polar form of a complex number. Fill in the missing lines so that the main function can print the correct polar form corresponding to a complex number taken as input in Cartesian form. [1+1+1+1+1=5 Marks]

```
typedef struct {
   float rel; float img;
} cartesian;

typedef struct {
   float mag; float theta;
} polar;

polar *cart2pol (cartesian a) {
  //Dynamically allocate memory

  polar  *b = (polar *) malloc(sizeof(polar));

  //Assign value for magnitude and theta in structure

   b→mag          = sqrt(a.rel*a.rel + a.img*a.img);

   b→theta         = atan(a.img/a.rel);

  return b;
}

int main() {
  cartesian x; polar * y; scanf ("%f%f", &x.rel, &x.img);

  y = cart2pol(x);                          //Call cart2pol function
  printf("%f %f", y→mag, y→theta);
  return 0;
}
```

b) The following function takes an array of students as input along with the length of the array and sorts the array in lexicographically ascending order of the **lastname**. Fill in the missing lines. You may use appropriate string library function(s). [1+1+1+1+1= 5 Marks]

```
typedef struct  {
   char firstname[30]; char lastname[30];
} fullname;

typedef struct {
   int roll; fullname name;
} student;

void selectionsort (student A[], int size) {
  int i, j, pos;
  student min, temp;
  for (i = 0; i < size - 1; i++) {
    pos = i;

    min = A[i]; //Save current minimum
    for (j = i + 1; j < size; ++j) {
      //Does lastname come before lastname in current minimum

      if (strcmp(A[j].name.lastname,min.name.lastname) < 0                    )
      {

        min = A[j];                  //Update current minimum
        pos = j;
      }
    }
   //Bring the smallest data to the current location of i

    temp = A[i];

    A[i] = A[pos];
    A[pos] = temp;
  }
}
```

QUESTION 8.

a) The following C recursive function **reverse_list** is used to reverse the contents of a linked list. It takes as parameter a pointer to a pointer to the first element of the list. Thus, starting from the head, if the list contains the following sequence of data: {56, 78, 20, 88, 80}, after the function is completed, the list will contain the following sequence of data: {80, 88, 20, 78, 56}. Fill in the missing lines.

[1+1+1+2=5 Marks]

```
struct node {
   int data; struct node *next;
};
void reverse_list (struct node **head) {
  struct node *previous   = NULL;

  struct node *current = *head;
  struct node *next;
  while (current != NULL) {

    next  = current→next;
    current→next = previous;
    previous = current;

    current = next;
  }
    *head = previous;
}
```

This question had a small mistake. The problem statement says “following C recursive function” whereas the code given is non-recursive. This may have created a confusion for some students, though chances are less as the code was quite clearly non-recursive and we have not received a single question or comments on the copy. Nevertheless, it is decided to give full marks to ALL students in this question.

b) The following recursive function **merge_sorted** takes as input the head pointers to two linked lists. In each linked list, the elements are in ascending order. The function returns the head pointer to a new linked list which contains all the elements of the two input lists in ascending order. Fill in the missing lines. Assume that the same definition of **struct node** as given in Question 8(a) is used here.

[1+1+1+1+1=5 Marks]

```
struct node *merge_sorted (struct node *a, struct node *b) {
  struct node *result = NULL;
  // Base cases

  if (a == NULL) return (b);

  else if (b==NULL) return (a);
 //Pick either from a or from b and recur

  if (a→data <= b→data                    ) {
    result = a;
    result→next = merge_sorted (a→next, b);
  }
  else {
    result = b;
    result→next = merge_sorted (a, b→next);
  }
  return(result);
}
```

QUESTION 9.

a) The following C program uses a stack of size 100 characters with the given definitions of **push()** and **pop()** to determine if an input string is a palindrome or not. A string is said to be a palindrome if it reads the same forward and backward. For example, **abcba** is a palindrome while **abcbc** is not. Assume that **push()** will never be called on a full stack and **pop()** will never be called on an empty stack. Fill in the missing lines. Use appropriate C string library function(s) if needed. [1+2+2+1= 6 Marks]

```
char stack[100];  int top = -1;
void push (char c) { stack[++top] = c; }
char pop() { return (stack[top--]); }

int main() {
  char input_string[30];     int i, count = 0, length;
  scanf("%s", input_string);

  length = strlen(input_string);
  //Push characters in stack

  for (i = 0; i < length; i++) push(input_string[i]);
  for (i = 0; i < length; i++) {
      //Compare with original string by popping from stack

       if (input_string[i] == pop())
            count++;
  }
  if (count == length)                           //Check condition for palindrome
    printf("Palindrome");
  else printf("Not palindrome");
  return 0;
}
```

b) What will be the displayed when the following C program is executed? [4 Marks]

```
typedef struct { int list[10]; int front, rear; } queue;
void enqueue(queue *q, int e)  {
  q→rear = (q→rear + 1);
  q→list[q→rear] = e;
}
void dequeue (queue *q) { q→front = (q→front + 1); }
int main() {
  int i = 0; int e = 4; queue *newqueue;
  newqueue = (queue *)malloc(sizeof(queue));
  newqueue→front = 0; newqueue→rear = -1;
  for (i = 0; i < 7; i++) { ++e; enqueue(newqueue,e); }
  for (i = 0; i < 3; i++) dequeue(newqueue);
  for (i = newqueue→front; i < newqueue→rear; i++) printf ("%d ", newqueue→list[i]);
  return 0;
}
```

```
8 9 10
```