Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

CS10003 Programming and Data Structures

Spring Semester, 2021-2022 **Marks = 30**

Short Test 2 **8-June-2022, 18:00 to 19:10**

INSTRUCTIONS

- You have 1 hour for writing and 10 minutes for submission.
- Write your answers on paper. Answers must be handwritten. Typed or written answers using an electronic device are not allowed.
- Write your name and roll number on each page. Write page numbers for each page.
- Scan all pages and collate. Create a single PDF file (of size $\leq$ 10 MB). You could also take pictures of different pages, combine them to make a single pdf file.
- The name of the files for parts should be <RollNumber>_st2. Ensure your roll number in the filename has only digits and uppercase characters.
- Upload your file. Make sure you click on the 'Turn in' button to submit your file.
- The said deadlines are strict after which no submissions will be allowed.
- We will not accept submission by any other means.

1. (a) Consider the structure below, where the rank field stores the class rank of a student. Fill in the blanks of the following code to print the class ranks of the students A and C. Your answer cannot contain the letters A and C.

```
struct student {
  int rank;
  struct student *ptr;
};

int main() {
  struct student A, B, C;
  A.rank = 1; B.rank = 5; C.rank = 10;
  A.ptr = &B; B.ptr = &C; C.ptr = &A;
  printf("%d,%d", __________ , ______________);          [A],[B]
}
```

2

Solution:

[A] ((B.ptr)->ptr)->rank
[B] (B.ptr)->rank

(b) Consider the following definition and assignment.

```
struct Person {
  int age;
  float weight;
} p, *t;
t = &p;
```

Suppose that you want to read age of p, and then print the same using scanf() and printf(). Fill in the blanks to do this. You can use t (and not p) for this purpose.

```
scanf("%d", __________ );          [A]
printf("%d", __________ );         [B]
```

2

Solution:

```
[A]  &(t -> age)
[B]  t -> age
```

2. A complex numer $z = x + iy$, is represented using 2 double-precision floating point values – one for the real part x and the other for the imaginary part y. The modulus of z denoted $|z|$ is given by $\sqrt{x^2 + y^2}$. If $z_1 = x_1 + iy_1$ and $z_2 = x_2 + iy_2 \neq 0$ are 2 complex numbers, then z_1/z_2 is given by

$$\frac{x_1x_2 + y_1y_2 + i(x_1y_2 - x_2y_1)}{x_2^2 + y_2^2}.$$

You have to calculate the following expression, where w_1, w_2 are two input complex numbers.

$$w = \left| \frac{w_1 - w_2}{w_1 + w_2} + 0.5 \right|$$

Following is an incomplete program that computes the value for w. Fill in the blanks to complete the program. You may assume that all necessary library files are included. 5

```
typedef struct {
  double x;
  double y;
}Complex;

Complex add(Complex z1, Complex z2) {
  Complex z;
  z.x = _________;                                    [A]
  z.y = _________;                                    [B]
  return (z);
}

Complex sub(Complex z1, Complex z2) {
// computes and returns z1-z2
}

double mod(Complex z) {
  return(______________________);                     [C]
}

Complex div(Complex z1, Complex z2) {                 // Divide z1 by z2
  Complex z;
  z.x = ________________________;                     [D]
  z.y = ________________________;  [E]
  return (z);
}

int main() {
  Complex w1;
  Complex w2;
  scanf("%lf %lf %lf %lf",&w1.x,&w1.y,&w2.x,&w2.y);
```

```
  Complex v = add(w1,w2);
  if(____________){                                    [F]
    printf("Error: division by 0.\n");
    return 0;
  }
  Complex w = _________________________;               [G]
  w.x += ___________________;                          [H]
  printf("|w| = %lf\n", mod(w));
  return 0;
}
```

Solution:

```
[A]     z_1.x + z_2.x
[B]     z_1.y + z_2.y
[C]     sqrt(z.x * z.x + z.y * z.y)
[D]     (z1.x*z2.x + z1.y*z2.y)/(mod(z2)*mod(z2))
[E]     (z1.y*z2.x - z1.x*z2.y)/(mod(z2)*mod(z2))
[F]     v.x == 0 && v.y == 0
[G]     div(sub(w1, w2), v)
[H]     0.5
```

3. Let A be a 2-d array of m rows and n columns, where $A[i][j]$ is the price per kg of the j-th item, $0 \leq j < n$, in the i-th month, $0 \leq i < m$. Let B be a 2-d array of n rows and m columns where $B[i][j]$ is the quantity of the i-th item, $0 \leq i < n$, purchased in the j-th month, $0 \leq j < m$. Compute the total expenditure on the purchase of all items in each of the m months in a 1-d array C of m elements. Assume $m, n \leq 25$. Write a C function `void expenditure(float A[][25], float B[][25], float C[], int m, int n)` that computes C. Assume that m, n, A, B are read from the user before being passed to the function. [7]

Solution:

```
void expenditure(float A[][25], float B[][25], float C[], int m, int n){
  int i,j;
  for (i=0;i<m;i++){
    C[i]=0;
    for (j=0;j<n;j++)                // for every item j compute the cost in month i
      C[i] = C[i] + A[i][j]*B[j][i]; // and accumulate for month i
  }
}
```

4. Write a C function called match which takes as argument a pointer (say p) to the start of a one-dimensional character string array (that eventually ends with a '\0') and returns the number of occurrences of lower-case vowels (including multiple occurrences) in the string pointed to by p. The function should use only one integer local variable (other than the function argument p) and no other variable. Write the function and the main program. In the main program, read the input, call the function and print the result. The input can have blanks in between but will not be more than 29 characters including blanks. The main program should only declare a single character array of size at most 30 and at most one integer variable. The user will give input in a single line and press the ENTER key. For example, if input string is "this is iit kharagpur", the function should return 7 and this should be printed in the main program. Do not use any library function other than standard input-ouput. [7]

Solution:

```
#include<stdio.h>
int match(char *p)
{
    int count;
    count = 0;
    while((*p)!= '\0')
    {
        if ((*p == 'a') || (*p == 'e') || (*p == 'i') || (*p == 'o') || (*p == 'u' ))
           count++;

        p++;
    }
    return count;
}
main()
{
    char word[20];
    int i;
    i = 0;
    do
    { scanf("%c", &word[i]);
      if (word[i] == '\n')
      {
          word[i] = '\0';
          break;
      }
      i++;
    } while(1);
    printf("Word read is %s \n", word);
    printf("Number of Vowels = %d \n", match(word));
}
```

5. Read a positive integer n and read in n sequences of integers. For each sequence, read in the number of elements of each sequence and then the elements of that sequence. All elements of the sequence are positive integers. Use dynamic allocation to optimally allocate and store the elements based on the number of elements. Then find out the largest number among all the sequences and print it. Write only one main program. Do not declare any static array. Declare and use at most 10 local variables, including pointers. Do not use any library or user-defined function other than standard input-output library. For example, suppose $n = 3$ and the number of elements for sequences 1, 2 and 3 are 3, 4 and 2, respectively. Sequence 1 is {4, 5, 8}, Sequence 2 is {9, 7, 13, 6} and Sequence 3 is {11, 6}, then the answer is 13. 7

Solution:

```
#include<stdio.h>
main()
{
    int **w;
    int n, m, i, j, k, found, largest;
    printf("Give number of Sequences: \n");
    scanf("%d", &n);
    printf("n = %d \n", n);
    w = (int **)malloc(n*sizeof (int *));
    for(i = 0; i < n; i ++)
    {
        printf("Give the length of %d-th Sequence\n", i+1);
        scanf("%d", &m);
        printf("%d-th m = %d \n", i+1, m);
        w[i] = (int *)malloc((m+1)*sizeof(int));
```

```
        w[i][0] = m;
        printf("Give all the %d elements of %d-th Sequence \n", m, i+1);
        for(j = 1; j <= m; j++) scanf("%d", &w[i][j]);
    }
    largest = 0;
    for (i = 0; i < n; i++)
         for (j = 1; j <= w[i][0]; j++){
               if (w[i][j] > largest)
                  largest = w[i][j];
         }
     printf("Largest = %d \n", largest);
}
```