INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Stamp / Signature of the Invigilator

EXAMINATION (End Semester)	**SEMESTER (Autumn)**

Roll Number										**Section**		**Name**	

Subject Number	C	S						**Subject Name**	*Programming and Data Structures*

Department / Center of the Student		**Additional sheets**	

Important Instructions and Guidelines for Students

1. You must occupy your seat as per the Examination Schedule/Sitting Plan.
2. Do not keep mobile phones or any similar electronic gadgets with you even in the switched off mode.
3. Loose papers, class notes, books or any such materials must not be in your possession, even if they are irrelevant to the subject you are taking examination.
4. Data book, codes, graph papers, relevant standard tables/charts or any other materials are allowed only when instructed by the paper-setter.
5. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, exchange of these items or any other papers (including question papers) is not permitted.
6. Write on both sides of the answer script and do not tear off any page. **Use last page(s) of the answer script for rough work.** Report to the invigilator if the answer script has torn or distorted page(s).
7. It is your responsibility to ensure that you have signed the Attendance Sheet. Keep your Admit Card/Identity Card on the desk for checking by the invigilator.
8. You may leave the examination hall for wash room or for drinking water for a very short period. Record your absence from the Examination Hall in the register provided. Smoking and the consumption of any kind of beverages are strictly prohibited inside the Examination Hall.
9. Do not leave the Examination Hall without submitting your answer script to the invigilator. **In any case, you are not allowed to take away the answer script with you.** After the completion of the examination, do not leave the seat until the invigilators collect all the answer scripts.
10. During the examination, either inside or outside the Examination Hall, gathering information from any kind of sources or exchanging information with others or any such attempt will be treated as '**unfair means**'. Do not adopt unfair means and do not indulge in unseemly behavior.

Violation of any of the above instructions may lead to severe punishment.

Signature of the Student

To be filled in by the examiner

Question Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Total**
Marks Obtained											

Marks obtained (in words)	**Signature of the Examiner**	**Signature of the Scrutineer**

1. (a) (2 marks) Write a program segment (c-statements) corresponding to the following: (1) declaring a file pointer (2) opening a file, with file name "abc.txt", for writing (file must be created if it does not exist); and writing the string "Hello World" to the file opened in (2).

(b) (2 marks) Write the output for the when the following main() function is executed.

```
void circular(int *a, int b, int *c) {
    int temp;
    temp = *a;
    *a = b;
    b = *c;
    *c = temp;
}
int main() {
    int x = 2, y = 3, z = 4;
    circular(&x,y,&z);
    printf("%d %d %d\n",x,y,z);
}
```

(c) (2 marks) Write the output for the when the following main() function is executed.

```
void foo(int x) {
    if( x > 5 ) foo(x-1);
    printf("%d ",x);
}
int main() {
    foo(9);
}
```

(d) (4 marks) Consider the sequence of numbers:

$$x_0 = 2$$
$$x_1 = 2 + \frac{1}{2}$$
$$x_2 = 2 + \frac{1}{2 + \frac{1}{2}}$$
$$x_3 = 2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}$$
$$\ldots$$

Write a function to compute the value of x_i given i.

2. (a) (5 marks) Write a function `insert()` to insert an integer `x` into a sorted array `A[]` (sorted in ascending order) congtaining `n` integers so that the array remains sorted after insertion. Return the length of the new array.

```
int insert (int x, int A[], int n) {
```

```
}
```

(b) (5 marks) Write a function `insertionsort()` that takes an array `A[]` containing `n` integers as input and uses insertion sort to sort the array by making calls to the function `insert()`.
Note: To get credit you must use calls to `insert()` appropriately.

```
void insertionsort (int A[], int n) {
```

```
}
```

3. Consider a program that stores 8-bit binary numbers in two's - complement representation, using null-terminated character arrays (c-strings). For example, `16` is stored as string ``00010000''. Complete the following functions to implement the program.

   (a) (5 marks) Write the function `bin2dec()`, which takes as input a character array `a`, encoding a 8-bit number in two's complement representation, and returns the decimal value for the encoded string as integer.

```
int bin2dec(char a[]) {
_____________________________________________________________
_____________________________________________________________
  if( a[0] == '1' ) { // negative number
    for(_____________________________________) {
_____________________________________________________________
_____________________________________________________________
    }
    ____________________________________________
  } else { // positive number
    for(_____________________________________) {
_____________________________________________________________
_____________________________________________________________
    }
  }
  return ret;
}
```

   (b) (5 marks) Write the function `add( )` which takes two strings `a` and `b`, containing two 8-bit binary numbers represented in two's complement form as input, and returns their sum as an 8-bit two's complement number in a null-terminated character array, `c`. You are not allowed to call any functions here. There will be no marks if you convert the numbers to integers, add them, and convert back to binary.

```
void add(char a[], char b[],char c[]) {
_____________________________________________________________
_____________________________________________________________
  // loop over the strings
  for(_____________________________________) {
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
  }
_____________________________________________________________
_____________________________________________________________
}
```

   (c) (1 mark) Complete the function `main( )` , which takes two strings `A` and `B` in two's complement format as input from the keyboard, prints the strings and their values (using `bin2dec()` function),

adds them (using `add()` function) and prints the result as binary string and corresponding decimal value.

```
int main() {
  char a[9], b[9],c[9];
  //read a and b from user and print them in both binary and decimal
```

_____

_____

_____

```
  //add and store the result in c
```

_____

```
  // print c in both binary and decimal
```

_____

_____

```
}
```

4. What will be printed when the following programs/ program segments execute? Write **only** the output that will be printed if the program is executed within the box.

   (a) (2 marks) Following program is executed as: `./a.out Hello World`

```
int main(int argc, char *argv[]) {
  printf("%d\n",argc);
}
```

   (b) (2 marks)

```
int a[10];
printf("%ld",(&a[7]-&a[3]));
```

   (c) (3 marks)

```
char s[3][5];
printf(%ld\n,(&s[2][2]-&s[1][1]));
```

(d) (5 marks)

```
int main()
{
  int array[10] = {4,3,5,6,1,2,9,8,5,4}, n=10, c, d, swap;

  for (c = 0 ; c < ( n - 1 ); c+=2)  {
    for (d = 0 ; d < n - c - 2; d++)    {
      if ( ( (d%2 == 0) && (array[d] > array[d+2]) ) ||
           ( (d%2 == 1) && (array[d] < array[d+2]) ) )
      {
        swap       = array[d];
        array[d]   = array[d+2];
        array[d+2] = swap;
      }
    }
  }

  for ( c = 0 ; c < n ; c++ )
     printf("%d ", array[c]);
  printf("\n" );
  return 0;
}
```

5. The following definition may be used for the rest of this question.

```
typdfef struct {
      float x;
      float y;
} POINT;
```

(a) (2 marks) Define a structure `struct Rectangle` to store a rectangle whose sides are parallel to the x and the y axis, in terms of coordinate of vertices of lower left and upper right vertices.

(b) (3 marks) Write a function `int inside (POINT p, struct Rectangle r)` that returns 1 if the point p is inside or on the rectangle r, and 0 otherwise.

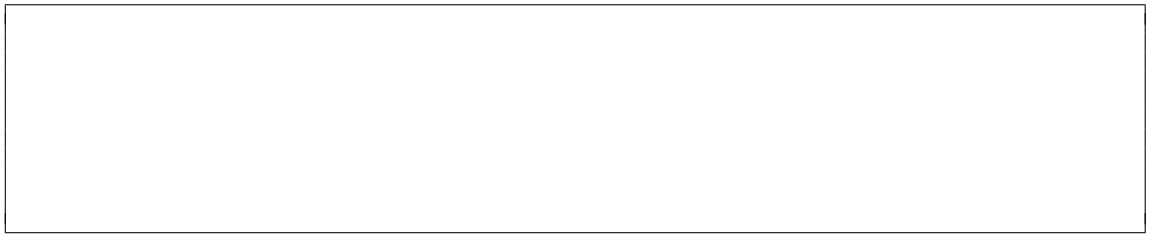
(c) (5 marks) Write a function that takes an array of rectangles of type (struct Rectabgle) and the number of rectangles as parameters, and returns a rectangle that is the smallest rectangle enclosing all the given rectangles.

6. Consider the following data type definition for representing elements of a linked list.

```
struct node {
     int data ;
     struct node * next;
}
```

(a) (5 marks) Write an `append()` function that takes two lists, `a` and `b`, appends `b` onto the end of `a`,and returns the head of the new list. Fill in the missing segments in the function below.

```
struct node * append (struct node *a, struct node * b) {
    struct node * new ;
    if (a==NULL)  {
```



```
         }
         else {
```

```
         }
         return a;
}
```

7. (5 marks) Consider an abstract data type of a queue containing integer data elements with the following specifications of the interface functions:

```
void enqueue (QUEUE *q, int element);
                        /* Insert an element in the queue */
int dequeue (QUEUE *q);
                        /* Remove an element from the queue */
queue *create();
                        /* Create a new queue */
int isempty (QUEUE *q);
                        /* Check if queue is empty */
int size (QUEUE *q);
                        /* Return the no. of elements in queue */
```

Given a Queue Q, write a function that will find the max element in the queue. You may only use queue operations such as eneque, dequeue, size etc.. No other data structure can be used other than queues. Queue must remain intact after finding the max.

```
int findmax (QUEUE *q)  {
int max,i;
// initialize max with the first element
```

_____

_____

```
// for all remaining elements
for (_____) {
```

_____

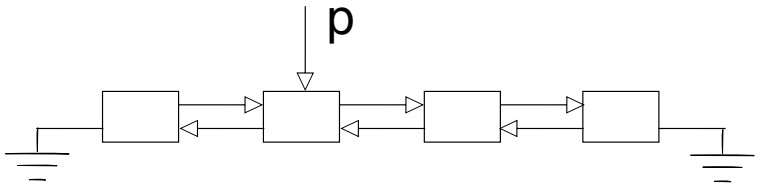
_____

_____

_____

```
    }
  return max;
}
```

8. (10 marks) Consider the following implementation of a priority queue (`Pqueue`), a list of integers which are sorted in descending order (highest first), using doubly linked lists. A doubly linted list is a linked list where one stores the pointers to both previous (`left`) and next (`right`) elements, shown in the figure below. The priority queue is used in the main program for sorting 10 randomly generated numbers. `rand()` function generates pseudo-random integers. Fill in parts of the insert function, so that the sorted property of priority queue is maintained. Note that you may receive pointer to any node of the list, not necessarily head node.




```
typedef struct pqueue {
  int item;
  struct _pqueue *left, *right;
} Pqueue;

int main() {
  Pqueue *p=NULL;
  int i;

  for(i=0;i<10;i++) {
    p=insert(p,rand()%200);
  }

  while(p->left != NULL) p=p->left;
  while(p->right != NULL) {
    printf("%d ", p->item);
    p=p->right;
  }
  printf("%d\n", p->item);
}


//insert a in p an return the pointer of the inserted node
Pqueue *insert(Pqueue *p, int a) {
  Pqueue *curr;
  //allocate memory and initialize all fields
  curr=(Pqueue *)malloc(sizeof(Pqueue));
  curr->item=a;
  curr->left=curr->right=NULL;
  //insert in empty list
  if(p==NULL) return curr;

  if(p->item > a ) {
  //insert to the left
    while( p->left != NULL) {
```

```
      if (p->item <= a) {
      // stop before reaching end
```

```
        return curr;
      }
    }
    //insert in left end
```

```
    return curr;
  } else {
  //insert to right
    while( p->right != NULL) {
```

```
      if(p->item >= a) {
      // stop before reaching end
```

```
        return curr;
      }
    }
    //insert in right end
```

```
    return curr;
  }
}
```

[Extra Page/ Rough Work]

[Extra Page/ Rough Work]