B. Sc. Engineering Examinations

(Structured Programming Language)

Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

SECTION – A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Write some differences between signed and unsigned data types. Can these qualifiers be applied to double or float?

(b) Write a simple program to find the size of different basic data types in C.

(c) Explain bitwise right shift operator. Write a program that takes an integer, x, as input and outputs how many '0' s are there at the end of the binary representation of the factorial of x. Assume 64 bit integer and factorial of x will not exceed integer boundary.

(d) Write a program to enter n elements in an array and find the second smallest number from the array. There cannot be any nested loops in your program.

2. (a) Write short notes on local, global and static variables. What are the features of static local and static global variables?

(b) Explain what will be the output of the following code?

```
# include <stdio.h>
void main ( )
{
    int  k = 10;
    switch    (k%2)
    {
        case  0  :  k+=2;
        case  1  :  k= 0;
        case  5  :  k=1;
    };
    printf  ("k=  %d", k);
}
```

(c) Write a C program to read n integer numbers in an array and split the array into two parts even and odd such that the even part contains all the even numbers and other is odd. Use dynamic memory allocation so that no memory is wasted.

e.g. Original array is 7,9,4,6,5,3,2,10,18

Odd array is 7,9,5,3

Even array is 4,6,2,10,18

Contd P/2

Contd ... Q. No. 2

(d) Rewrite the following code with do-while loops only.

```
for (i = 1; i<=5; i++)      {
    for (j=1; j<=i; j ++)
        printf ("%i", i);
    printf ("\n");
}
```

3. (a) Define Macro and write a macro in "C" to swap two data items.

   (b) What is the typedef declaration? Give suitable examples.

   (c) An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself (e.g. $153=1^3+5^3+3^3=153$). Write a C language program to print all Armstrong numbers between 100 and 999.

   (d) Write a function **char *reverseword(char *s)**, which takes a string as input, detects each word and reverses each word leaving blanks intact. The function should return the resultant string's pointer.

4. (a) Without using if/else or '?' operator, write a function **int PositiveSubtract(int x, int y)** which computes x - y when, x > = y, otherwise returns 0.

   (b) Write a program to find whether N is a super-prime or not.
   Input: 7331 (is super prime) Input: 4550 (is not super prime)
   Hint: 7331 is super-prime because, 7331, 733, 73 and 7 are primes

   (c) Write a program to sort all the elements in a 4×4 matrix rowwise.
   E.g.

Input: $\begin{bmatrix} 4 & 3 & 2 & 1 \\ 4 & 3 & 3 & 1 \\ 4 & 3 & 2 & 1 \\ 4 & 3 & 2 & 2 \end{bmatrix}$ Output $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 3 & 4 \\ 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \end{bmatrix}$

SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

5. (a) Write a program that takes from user a file name as input and counts the total number of lines in the file. Lines are defined as ending with a newline character. The output of the program should be the line count. Your program should also check for any possible errors that can happen, such as file is not found in the disk.

Contd P/3

Contd ... Q. No. 5

(b) Consider the following structure definition for storing the nodes of a linked list:

```
struct  Node
{
    int val;
    struct Node * next;
};
```

Now, write the following functions assuming the above mentioned structure definition:

(i) Write an insert function that will insert elements in the linked list maintaining **ascending sorted order**. The prototype of the insert function should be as follows:
*struct Node * insert(struct Node * head, int val)*;
The function receives two arguments. The first one is the head (start) pointer of the linked list. The next argument is the integer value that is to be inserted. The function first searches the existing list to find the appropriate place where the value should be stored. The function then creates memory for the new value and then inserts it. The function returns the new head pointer (head pointer may change after insertion) of the linked list.

(ii) Write a recursive function that will print the linked list in **reverse order**. The function will receive the head (start) pointer of the linked list as its arguments and recursively prints the all elements of the list in reverse order. The prototype of the recursive function should be as follows:
*void print(struct Node * elem)*;

6. (a) What are the advantages and disadvantages of using macro over functions?

(b) Write the definition of a macro CUBE(x). The macro should compute the cube of x. Write the macro so that the following statements evaluate correctly:

```
int x, y, z;
x = CUBE(–5);
y = CUBE(4–1);
z = !CUBE(3);
```

(c) Briefly show the uses of the following library functions with example programs:

(i) *malloc* function defined in stdlib.h
(ii) *atoi* function defined in stdlib.h
(iii) *Strcmp* function defined in string.h

(d) Write a function strrindex(s,t), which returns the position of the rightmost occurrence of string t in string s, or –1 if there is no such occurrence. The prototype of this function is given below:

int strrindex(char s[], char t[]);
Sample input: s="aabcaabc", t="aa", Sample output: 4
Sample input: s="aabcacacd", t="bc", Sample output: 2
Sample input: s="abcabcabc", t="abcd", Sample output: –1

Contd P/4

7. (a) The following function is supposed to allocate memory for an integer array:

```
#include<stdlib.h>
void allocate(int *t,int size)
{
    t = (int*)malloc(sizeof(int)*size);
}
int main()
{
    int * p;
    allocate(p, 100);
}
```

Does the allocate function written above successfully allocate memory for variable p in main function? Justify your answer.

(b) Write a program that will take a number of command line arguments. All arguments will be integer numbers. Your program will find the median and mode of those numbers. To find the median, the numbers have to be listed in numerically sorted order. Then the middle element is the median. For example, if the list is {4, 1, 1, 10, 3, 5, 7}, then first sorting in ascending order, we find the list as {1, 1, 3, 4, 5, 7, 10}. The median will be the 4-th element which is 4. Note that, if the number of elements in the list is even, then the average of two middle elements will be the median.

The mode is the number that is repeated more often than any other, so in the above list 1 is the mode. If there is no such number which appears maximum times, your program should print "No Mode".

Sample input: {4,1,1,10,3,5,7} sample output: mean=4, mode=1
Sample input: {2,1,4,7} sample output: mean=3, mode=No Mode.

(c) What are the advantages and disadvantages of using linked list over arrays?

8. (a) Consider the following definition:

```
struct Bookinfo
{
//define a field for title
//declare a field for author's name
//declare a field for price
};
```

i) Complete the above structure definition by declaring members/fields for storing book titles (at most 99 character long), authors name (at most 49 character long) and price of book (a real number).

(ii) Using the above structure, write a main program that will receive from user information (title, authorname, price) of ten books and store them in a file "bookinfo.bin". Note that, you have to use the above structure and fwrite function for storing data in the file. The prototype of the fwrite function is given below:

*size_t fwrite (void * Buffer, size_t Size, size_t Count, FILE * Stream);*

Contd P/5

Contd ... Q. No. 8(a)

(iii) Now, write another program that will ask user for a book title. The program will then search the "bookinfo.bin" file and retrieve the price of the book. In this case, **you have to use the fread function using the structure defined above for retrieving records from the file**. The program will print the price of the book. If the title is not found in the file, then program should print a line "Sorry, we do not have the book, Now".
The prototype of the fread function is given below:
*size_t fread (void * Buffer, size_t Size, size_t Count, FILE * Stream);*

(b) Declare an array of 5 pointers to characters. Allocate memory for these 5 pointers to store the strings "Bangladesh", "India", "SriLanka", "Nepal", and "Bhutan". You should allocate as much memory as required for storing the strings. Do not allocate extra memory. After allocating memory, show how you will store the strings in the array.

B. Sc. Engineering Examinations

(Structured Programming Language)

Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

All the questions in this section are related to C Programming.

1. (a) A C program contains the following array declarations:

```
char text[80];
```

Suppose that the following string has been assigned to text

```
Programming with C can be a challenging creative activity.
```

Show the output resulting from the following *printf()* statements.

(i) printf("%18s", text); (ii) printf("%.18s", text);

(iii) printf("%18.7s", text); (iv) printf("%–18.7s", text);

Use␣ to show each blank position.

(b) Write an interactive C program that will convert a date, entered in the form mm-dd-yy (example: 4-12-13) into an integer that indicates the number of days beyond January 1, 2000. If the year does not extend beyond 2199 (i.e., if yy <= 99), we can make use of the following relationships:

(i) The day of the current year can be determined approximately as

day = (int) (30.42) * (mm – 1)) + dd

(ii) if mm = = 2 (February), *increase* the value of day by 1.

(iii) if mm > 2 and mm < 8 (March, April, May, June or July), *decrease* the value of day by 1.

(iv) if yy%4 = = 0 and mm > 2 (leap year), *increase* the value of day by 1.

(v) *Increase* the value of day by 1461 for each full 4-year cycle beyond 1-1-2000.

(vi) *Increase* day by 365 for each additional full year beyond the completion of the last full 4-year cycle, then add 1 (for the most recent leap year).

(c) A table contains *n* unique data elements sorted in ascending order. To find a data element, the table is first split into blocks of *m* data elements, and then a linear search is performed on the last data element of each block to find the block containing the target data record. If the number of elements in the selected block is less than *m*, then a linear search is performed on the selected block to find the target data element; otherwise the selected block is further split into blocks of *m* data elements and again a linear search is performed on the last data elements recursively. Write a recursive C function that is to be used to implement the searching of an element from a list of integer data using the stated algorithm.

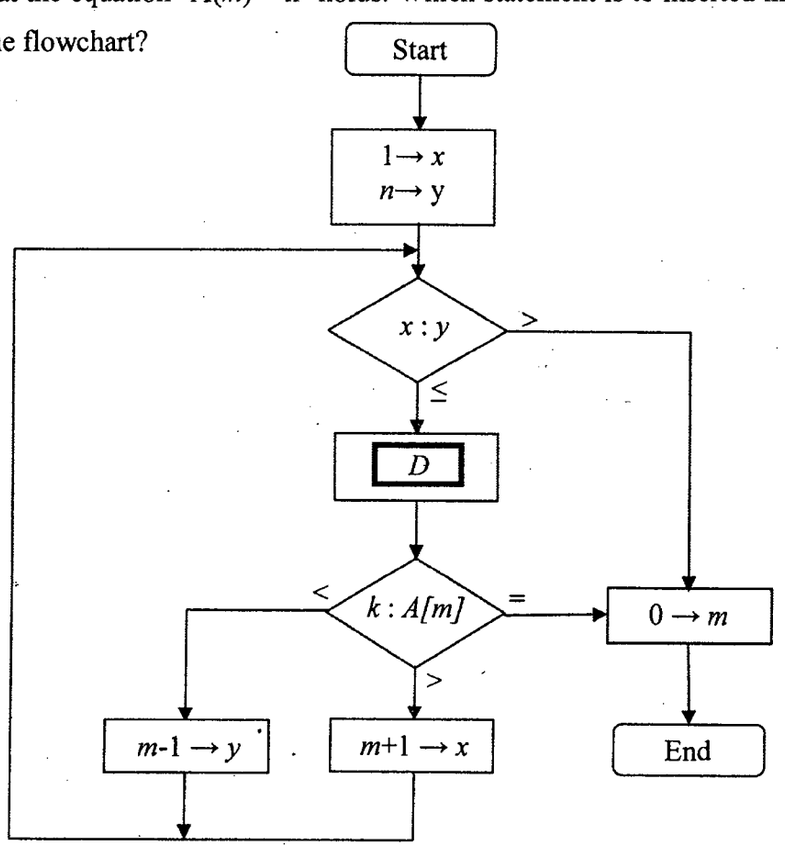
(d) What is escape sequence? What is the purpose of using escape sequence in C programming?

Contd P/2

2. (a) Describe the output of the following program:

```
#include <stdio.h>
int a = 0, b = 1;
int funct1(int a);
int func2(int b);
main(){
        int count;
        for(count = 1; count <= 5; ++count){
                b += funct1(a + 1) + 1;
                printf("%d ", b);
        }
}
funct1(int a){
        b = funct2(a + 1) + 1;
        return(b);
}
func2(int a){
        return(b + a);
}
```

(b) The flowchart below shows a binary search algorithm to find the index m of the array element A(m), such that the equation "A(m) = k" holds, from the array elements A(1), A(2), ... A(n) already sorted in ascending order. In case of "m = 0" at the end, there is no element such that the equation "A(m) = k" holds. Which statement is to inserted in the process box D in the flowchart?

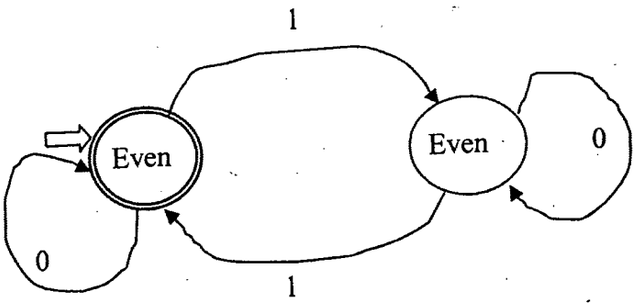


Write an iterative C function for implementing the flowchart stated in the figure.

Contd P/3

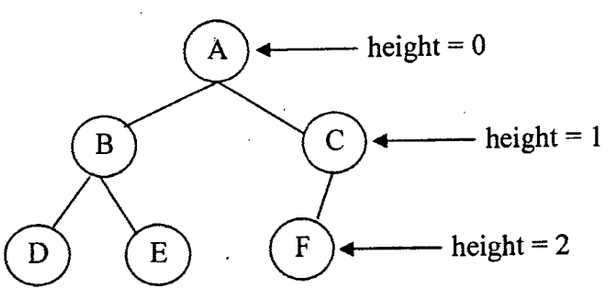
Contd ... Q. No. 2

(c) The figure below shows the state transition diagram of an automaton that accepts bit strings with even numbers of 1s. The double circle marked with "Even" represents the accepted state.



Binary digits, i.e., 1 or 0 are already stored in the array element *A[n]*. Assume that both *A[...]* and *n* are declared in the calling function and are propagated to the called function as arguments. Write a C function to determine whether the binary number stored in the array element contains "Even" number of 1s. The algorithm of the function must follow the state transition diagram shown in the figure.

3. (a) A binary search tree can be represented using a 1-D array data structure where the root of the tree is placed in the location indexed by 1 shown in the figure. In such a tree, there is no wastage of memory location for a complete binary search tree but some locations are kept blank if the tree is incomplete. Moreover, non-existence of a child is represented by –1.



Index→	0	1	2	3	4	5	6	7	8	9	...
A:		A	B	C	D	E	F	–1	–1	–1	...

(i) Write a C function that reads integers from the standard input device and place them in *A[n]* based on the principle of 1-D array based binary search tree representation. Assume that both *A[...]* and *n* are declared globally.

(ii) Write a C function to calculate the height of an element in the tree, when the height of the root is considered as zero (hint: the height of node D as shown in the figure is 2, the height of a non-existence node is –1).

Contd P/4

Contd ... Q. No. 3

(b) Write down some characteristics that are to be followed in writing a good computer program.

(c) Explain what happens when the following statement is executed.

if (abs(x) < $xmin$)x = (x > 0) ? $xmin$: $-xmin$;

Is this a compound statement? Is a compound statement embedded within this statement?

4. (a) Write a C function to delete an integer element from a sorted array. The array is left tidy and sorted.

(b) What is function prototype? Explain with example. Write a C function named "swap" which inter-exchanges the values of x and y and does not need to declare any variable within its body.

(c) Write a recursive C function to solve the following Legendre Polynomials

$$P_n = [(n\text{-}1)/n]xP_{n-1} - [(n\text{-}1)/n]P_{n-2}$$

where $P_0 = 1$, $P_1 = x$, $n = 1, 2, \ldots$ and x is any floating point number between –1 and 1. Describe the output of the function for $n = 5$ and $x = 0.5$.

SECTION – B

There are **FOUR** questions in this section. Answer any **THREE**.

5. (a) What is doubly linked list? Define the structure of a node in a doubly linked list where each node needs to store two integer variables and one character variable.

(b) Suppose you have a doubly linked list. Assume that each node of the list contains an integer variable named 'data'. Also assume that the two pointers are 'next' and 'prev' (apply common sense in using the two pointers). Assume that you have used **typedef** to define the type of the structure used in the linked list as 'DLL'. Now you are given a function named "findNode()" as defined below. Assume that 'pHead' is used to pass the pointer pointing to the start of the doubly linked list.

```
DLL * findNode(DLL *pHead, int target)
{
        DLL *pCur;
        /*search the nodes in a linked list*/
        pCur = pHead;
        /* search until the target value is found or the end of the list is reached*/
        while (pCur != NULL && pCur -> data != target)
                pCur = pCur -> next;
        if (pCur != NULL)
                return pCur->next;
        else
                return NULL;
}
```

Contd P/5

Contd ... Q. No. 5(b)

Now you need to write two functions the prototypes of which are given below:

```
int deleteNode(DLL *pHead, int target);
int insertNode(DLL *pHead, int target, int newData);
```

Both the functions "deleteNode()" and "insertNode()" first need to find the node which has the value target in its 'data' variable. Then "deleteNode()" deletes that node and "insertNode()" inserts a new node after that node. The new node should have 'newData' in its 'data' variable. If there exists no node in the linked list having the value 'target' in its 'data' variable, then "deleleNode()" does not delete anything and returns 0; otherwise it returns 1. If there exists no node in the linked list having the value 'target' in its 'data' variable, then "insertNode()" does not insert anything and returns 0; otherwise it returns 1. Both "deleteNode()" and "insertNode()" **MUST** use the given function "findNode()" to find the node which has the value 'target' in its 'data' variable and need to work with the pointer return by "findNode()". You CAN NOT change the function "findNode()" and hence you need to carefully understand what it really returns at different circumstances.

(c) Now assume that the doubly linked list is arranged in ascending order of the values in the 'data' variable. Now, can you write another function "insertNode1()", such that if "insertNode()" returns 0, then it searches and finds the node that has the **predecessor** value of 'target' in its data field and then inserts the new node after that node. Predecessor of a value is the highest value that is less than it, i.e., in sorted order (ascending), the predecessor of a value will always occupy the previous position of it.

6. (a) Write down an appropriate definition of a structure which has a float variable, an integer variable and a character variable. What will be the storage requirement of a variable of this type of structure? What will be the storage requirement if you use a union instead of a structure?

(b) Discuss the use of a tagged union. Suppose you need to handle two types of students. One type of student is given marks in percentage (e.g., 90%, 89.9% etc.) and the other type of students are given letter grades (e.g., A, A+ etc.). Use the idea of a tagged union to write a program for such a scenario. Your program will take input of 5 students from a user and put the data in an array. Then it will show the output.

(c) Your friend has written the following program which gives no compiler error. However, your teacher says that there is/are problem(s) in the code. Identify the problem and suggest possible corrections.

Contd P/6

Contd ... Q. No. 6(c)

```
struct ADate {
        int month;
        int day;
        int year;
};
int main(void)
{
        struct ADate *d;
        d->month = 3;
        d->day = 9;
        d->year = 2014;
        return 0;
}
```

(d) Deduce the storage requirement of the variable 'flag' below. Mention the range of values allowed for f1 and f2.

```
struct Flags {
  int      f1:3;
  unsigned int f2:1;
  unsigned int f3:30;
} flag;
```

7. (a) Write a program named 'myCopy' that will copy the content of one file (source file) to another file (destination file). The user may provide upto two file names following the name of the exe file (in this case, 'myCopy.exe'). If two file names are provided, the first file will be used as the source file and the second will be used as the destination file. If one file name is provided, then it will be used as the source file. The program then will have to ask for the destination file from the user. If no file name is provided then the program will ask for both the source file and the destination file. In case of any error in opening either the source or destination files or both, the user must be informed and the program again asks for another file.

(b) Suppose you want to open an existing file for both read and write operations. Your friend suggested that you may use either 'r+', 'w+', or 'a+' as the mode for file opening. Do you agree with him? Justify your answer.

(c) What could be the problem of the following macro definition?

```
#define SQUARE(x) x * x
```

Contd P/7

8. (a) Consider the following code fragment. Assume that x and p are correctly declared as an integer and a pointer to integer respectively. Explain, what does the following loop do?

```
...
p = &x;
x = 0;
while (*p = = x) {
        *p = *p + 1;
}
```

(b) Suppose your friend wrote the following function. He needs to swap the values of two variables and calls the swap() function from main() passing two variables (say, a and b). Unfortunately, he realized that the swap() function is not working correctly. Could you explain why it is not working? Can you correct the swap() function to make it work correctly?

```
void swap(int x, int y)
{
        int tem;
        temp = x;
        x = y;
        y = temp;
}
```

(c) Consider the following program:

```
#include<stdio.h>
int main(void)
{
        char *pChur;
        int * pInt;
        char aChar[8]= {0x78, 0x56, 0x34, 0x12, 0x12, 0x34, 0x56, 0x78};
        printf("%X\n", aChar;/*Output Line #1*/
        pChar = aChar;
        pInt = (int *)aChar;
        printf("%X\n",pChar);/*Output Line #2*/
        printf("%X\n",*pChar);/*Output Line #3*/
        pChar++;
        printf("%X\n",pChar);/*Output Line #4*/
        printf("%X\n",*pChar);/*Output Line #5*/
        printf("%X\n",pInt);/*Output Line #6*/
        printf("%X\n",*pInt);/*Output Line #7*/
        pInt++;
        printf("%X\n",pInt);/*Output Line #8*/
        printf("%X\n",*pInt);/*Output Line #9*/
        return 0;
}
```

Contd P/8

Contd ... Q. No. 8(c)

After executing the program, Output Line #1 gives 22FF24. What will be written in Output Line #2 to Output Line #9?

(d) Explain the following complicated declaration:

```
char (*(*x())[5])();
```

(e) The WndProc callback function of a windows program written in C is given below:

```
LRESULT CALLBACK WndProc(HWND hwnd, UNIT msg, WPARAM wParam,
LPARAM lParam)
{
    switch(msg)
    {
        case WM_LBUTTONDOWN:
        /*this is the corresponding message for left mouse button click*/
        {
            char param1 [10];
            char param2 [10];
            getUserInput(param1);
            getUserInput(param2);
            MessageBox(hwnd, param1, "The first parameter is:", MB_OK|
            MB_ICONINFORMATION);
            MessageBox(hwnd, param2, "The second parameter is:", MB_OK|
            MB_ICONINFORMATION);
        }
        break;
        case WM_CLOSE:
            DestroyWindow(hwnd);
        break;
        case WM_DESTROY:
            PostQuitMessage(0);
        break;
        default:
            return DefWindowProc(hwnd, msg, wParam, lParam);
    }
    return 0;
}
```

The procedure above uses the following API:

void getUserInput(char *ptr)	It takes input from the user and put it as a string in the memory pointed to by 'ptr'.

Now explain what happens when left mouse button is clicked? Update WndProc() so that when the left mouse button is clicked the summation of the two parameters are shown in a message box. Assume that the users would provide numeric input. For example, if the user provides "123" and "456", a message box would show: "579".

B. Sc. Engineering Examinations

(Structured Programming Language)

Time : 3 Hours

The figures in the margin indicate full marks.

USE SEPARATE SCRIPTS FOR EACH SECTION

SECTION – A

There are **FOUR** questions in this Section. Answer any **THREE**.

1. (a) Write a structure definition called *date* that contains three bit fields, called *day*, *month* and *year*, whose widths are 5 bits, 4 bits and 7 bits, respectively. Also write a union definition called *identity* in which a *date* type variable named *birthdate* share the same memory location with an integer type variable called *slnumber*. Use the keyword *typedef*, if necessary. How many decimal digits are possible to preserve for year in the memory? Write a C function that reads birthdate as an integer, extract day, month and year from it and prints the date in the form of *dd-mm-yy*.

   (b) (i) Write preprocessor directives for the following situations:

   If *flag* has a value of 0, define the symbolic constant COLOR to have a value of 1. Otherwise, if the value of *flag* is less that 3, define COLOR to have a value of 2; and if the value of *flag* equals or exceeds 3, define COLOR to have a value of 3.

   (ii) Replace the following code block by defining a macro called display (i) and using a *for* loop-

```
printf("x1 = %f\n", x1);
printf("x2 = %f\n", x2);
printf("x3 = %f\n", x3);
---------------------------
printf("x10 = %f\n", x10);
```

   (c) Write a C function that has the following function prototype-

   double avgPrice(const char *str, ...);

   The function is able to calculate the average price of all books listed in a string containing any number of <book name, price> pairs. A sample function call is as follows-

```
avg = avgPrice("book list", "mathemetics", 12.75, "programming with C", 20.75,
"physics", 23.20, NULL);
```

2. (a) Rewrite the following declaration and allocate equal amount of memory using dynamic memory allocation-

   int matrix [NoRow][NoCol];

Contd P/2

Contd ... Q. No. 2

(b) Consider the following declarations:

```
typedef struct{
        int  month;
        int  day;
        int  year;
}date;
typedef struct xx{
        char   empID[10];
        char   name[40];
        char   street[80];
        char   city[20]
        date   birthdate;
        int    salary;
        sturct xx  *next;
} record;
```

Assume that a link list has already been created based on the above structure declaration. The first element is named as "start" and the last element is NULL.

(i) Write a C function to display empID, name and city information of all employees in a tabular form whose salary is higher than or equal to 20000.

(ii) Write a C function to delete all employees from the list whose city = "Khulna".

(c) Write a C function with the following function prototype-

```
int strpos(const char *str, char ch);
```

The function will find the first occurrence of the character *ch* in the given string *str* and return the position index of the character in the string. The first position of the string is known as the position 0. If the character is not present in the string the function will return - 1.

3. (a) Explain the purpose and semantic meaning of the following C function using necessary figures:

```
int main(int argc, char  **argv){
    while (--argc> 0 && (*++argv)[0] == '-')
        while (c = *++argv[0]) printf("%c\n",c);
}
```

(b) A C program contains the following statements.

```
char u, v = 'A';
char *pu,  *pv = &v;
```

Contd P/3

Contd ... Q. No. 3(b)

```
*pv = v + 1;
u = *pv + 1;
pu = pv;
pu = &u;
```

Suppose each character occupies 1 byte of memory. If the value assigned to *u* is stored in (hexadecimal) address F8C and the value assigned to *v* is stored in address F8D, then

(i) Which statement is invalid?

(ii) What value is represented by &v?

(iii) What value is assigned to pv?

(iv) What value is represented by *pv?

(v) What value is represented by *pu?

(c) A C program contains following statements.

```
typedef struct tt{
    int value;
    struct tt *lchild;
    struct tt *rchild;
} node;
node *tree;
```

The value *tree* is a binary search tree. Therefore, for each node, all the values in the left subtree of the node is less than the value in the node and all the values in the right subtree is higher than the value in the node.

Write a C function of the following prototype that will insert a node into the tree.

```
void insertnode( node *n);
```

4. (a) The Towers of Hanoi is a well-known children's game, played with three poles and a number of different-sized disks. Each disk has a hole in the center; allowing it to be stacked around any of the poles. Initially, the disks are stacked on the leftmost pole in the order of decreasing size, i.e., the largest on the bottom and the smallest on the top. The object of the game is to transfer the disks from leftmost pole to the rightmost pole, without ever placing a largest disk on top of a smaller disk. Only one disk may be moved at a time, and each disk must always be placed around one of the poles. The general strategy is to consider one of the poles to be origin, and another to be the destination.

The third pole will be used for intermediate storage, thus allowing the disks to be moved without placing a larger disk over a smaller one. Write a recursive C function that will solve the above mentioned problem and describe the output for $n = 4$.

Also calculate the number of movements required to move n disks from leftmost pole to the rightmost pole.

Contd P/4

Contd ... Q. No. 6

(b) Write a C program that accepts two filenames as command line parameters and appends the second file to the first. If the user forgets to provide one or both filenames while running the program from the command line, the program will ask for and read the filename(s) accordingly.

(c) Discuss the formats and differences between read() and fread() functions.

SECTION – B

There are **FOUR** questions in this Section. Answer any **THREE**.

5. (a) Consider the following code fragment.

```
int  i = 'c';
```

Are we allowed to assign a character constant to an *int* variable? Please first answer yes or no, and then explain your reasoning.

(b) The *ceiling* of a *float* value is the smallest integer greater than or equal to it. On the contrary, the *flooring* of a float value is the largest integer smaller than or equal to it. For example, the ceiling of 3.12 is 4 and its flooring is 3. Write a C program, that takes a float value as input and prints its ceiling and flooring values. You are not allowed to use any library function except *input/output* related ones. [*Hint: Think about type conversion*]

(c) Write down a C program to calculate $\sqrt{|x|}$ where x is a double value. You can not use any library function.

(d) The speed limit on a highway is 100 km/h. If someone is caught speeding, they are given a fine that depends on their speed, as well as whether they have had a previous speeding ticket. The following table summarizes the fine that will be issued.

speed (km/h)	Previous ticket?	Fine amount
100 or less	not applicable	$0
101-120	no	$50
>120	no	$80
>100	yes	$150

Write a program that asks for an integer traveling speed (in km/h), and a **y** or **n** response to whether they have a previous ticket. Your program should then output the amount of the fine ($0, $50, $80, or $150).

6. (a) When we say $a = b$, a is assigned the value of b, and the expression evaluates to the value of a after the assignment. Given this, carefully explain when the *printif* in the following code would execute.

```
if(a = b) printf("The if statement is executing.");
```

(b) Briefly describe what is meant by *short-circuit evaluation,* and give a code fragment whose behavior depends on it.

Contd P/5

Contd ... Q. No. 6

(c) A *Mersenne Number* is a positive integer that is one less than a power of two. For example, the *n*-th Mersenne number, M_n, is:

$$M_n = 2^n - 1$$

A prime number is a positive integer that has exactly two distinct positive integer divisors, namely 1 and itself. The number 1 is, by definition, not a prime number. A Mersenne prime is a Mersenne number that is prime. The number 3 is the smallest Mersenne prime, because 3 is a prime number, and $2^2 - 1 = 3$. As of February 2014, 48 Mersenne primes are known. Write a complete C program that finds and prints the 5 smallest Mersenne prime, each by itself on a line. You may use any C math library functions in your solution.

(d) The number 153 has the property it is equal to the sum of the cubes of its digits:

$$1^3 + 5^3 + 3^3 = 153$$

Write a complete program that will find and print all the three-digit natural numbers that have this property.

7. (a) Rewrite the following statements so that a **do-while** loop is used instead of the **while** loop. The results from executing the statements must remain the same.

```
int sum = 0, num;
scanf("%d", &num);
while (num> =0)
{
    sum + = num;
    scanf("%d", &num);
}
```

(b) Write a complete C program that reads an integer *n* and computes and prints the term a_n where $a_1 = 1$, $a_2 = 2$, $a_3 = 3$, and for $k > 3$: $a_k = a_{k-2} + 2 \times a_{k-3}$. Assume n > 0. Below is the output from a sample run of the program:

Enter n : 5 [enter]

a_5 is 7

In your solution, you may use loop, but you are not allowed to use arrays, nested loop or recursion.

(c) Write a C program that prints **YES** if an integer entered by a user has symmetric bits around the middle and **NO** otherwise. An integer of size *n* bits is symmetric if the bit at position 0 is equal to the bit at position *n* - 1, bit a position 1 is equal to the bit at position *n* -2, and so on. For example, if the integer contains the following bit pattern then it is symmetric:

0100 1010 1110 0100 0010 0111 0101 0010

Contd ... Q. No. 7

On the other hand, the integer containing the following bit pattern in not symmetric:

0111 0010 1010 1111 0010 0111 0101 0010

Use bit-wise operators to solve the problem.

8. (a) Write a function **int allRepaeated (int i)** that returns 1 if and only if positive integer i consists only of pairs of repeated digits and 0 otherwise. The following table gives several examples of what the function should return.

Function call	**Returns**
allRepeated (8)	0
allRepeated (11)	1
allRepeated (2233)	1
allRepeated (22332)	0
allRepeated (222)	0
allRepeated (2323)	0

(b) Write a C function named **int countDistinct (int list[], int listLength)** that has two parameters. The parameter *list* is an integer array already sorted in ascending (nondecreasing) order, and the parameter *listLength* gives the number of elements in *list*. The function counts and returns the number of distinct elements in *list*. For example, if list is {1, 1, 3, 3, 3, 3, 6, 8, 8, 9, 9, 9, 9, 9}, the function returns 5, as list has 5 distinct elements.

(c) State two limitations of a **switch** statement over **if-else** statement with example.

B. Sc. Engineering Examinations

(Structured Programming Language)

Time : 3 Hours

USE SEPARATE SCRIPTS FOR EACH SECTION

The figures in the margin indicate full marks.

SECTION – A

There are **FOUR** questions in this section. Answer any **THREE**.

1. An analog clock has only two hands - hour and minute. The input for the program will be a set of 24 hour format time. For the given input the program should output all the input time and the angle created for that particular time in ascending order.

   The input for the program will be given in the following order. The first line will contain an integer **N** where $1 \le N \le 100$. The next N lines will contain two integers $hh(0 \le hh \le 23)$ and $mm(0 \le mm \le 59)$ separated by a blank space indicating hour and minute of a particular time in 24 hours format. For the input set the program should provide N lines as output. Each line should contain three integers separated by single blank space. The first two numbers are the hour and minute of the input set and the last number (rounding to integer) is the angle between the hour and minute hands. These three numbers should appear in the ascending order of the angle between the hour and minute hands. If there are times with the same angle then the earlier time should appear first (00:00 is the earliest time and 23:59 is the latest time)

   Sample input and output for the program may be:

Sample input	Sample output
4	0 0 0
3 0	23 59 6
9 0	3 0 90
23 59	9 0 90
0 0	

   In the program sections there is one user defined structure named **Time**. This structure has three integer variables ***hh, mm,*** and ***angle;*** and one long integer value ***sortingKey.*** ***Hh*** and ***mm*** are used to store the hour value and minute value of the time. Variable ***angle*** holds the value of the angle $\le 180^{\circ}$ (acute and obtuse angle only) created by the hour and minute hands of the clock. Variable ***sortingKey*** holds a value that is used to sort the list of time according to the problem statement.

   The program contains a function with prototype ***void SortTime(int i)***. The description of the function is as follows–

Contd P/2

Contd... Q. No. 1

Void SortTime(int i). This function basically combines three different numbers (***angle, hh*** and ***mm***) to get a value and set that value to the ***sortingKey*** variable of a particular element having index ***i***. For example: if the ***i***th element of the array ***t[]*** has ***hh = 16, mm = 55*** and ***angle = 177*** the function will combine the numbers ***177, 16*** (from ***hh***) and ***55*** (from ***mm***) to get the value ***1771655*** and then sets this value to the variable ***sortingkey*** of the ***i***th element. When the total list is sorted in the ascending order of this ***sortingKey*** the list is eventually sorted in the ascending order of ***angle*** and then by time.

```
void main()
      {
      long n,i,j,hour,totalMin;
      float hAngle,mAngle,angle;
      scanf("%ld",&n);
      for(i=0;i<n;i++){
         scanf("%ld%ld",&t[i].hh,&t[i].mm);
         hour=t[i].hh;
         if( [ A ] ){
           hour=t[i].hh-12;
         }
         totalMin = [ B ]
         hAngle=round(totalMin*0.5);
         mAngle=t[i].mm*6;
         if(hAngle>mAngle){
           t[i].angle=hAngle-mAngle;
         }
         else{
           t[i].angle=mAngle-hAngle;
         }
         if(t[i].angle>180){
            [ C ]
         }
         SortTime(i);
      }
      Sort((void *)&t, n, sizeof(Time));
      for(i=0;i<n;i++){
         printf("%d %d %ld\n",t[i].hh,t[i].mm,t[i].angle);
      }
}
```

(a) Fill up the blank A, B and C.

(b) Write down the structure defined in the problem description with 5 bits for hour, 6 bit for minute, 8 bit for angle and required minimum number of bits for ***sortingKey***. Put ***sortingKey*** in a new word. Use ***typedef*** to redefine it to a name of ***Time***.

What is the value of ***sizeof(Time)***?

(c) Write down the codes for the ***Sort()*** function which is referenced in the main function.

Contd P/3

2. The function ***putInt(int var)*** accepts one argument and displays the value of ***var*** on monitor screen. There are total three functions of similar kind that displays three data types named integer, float and character. The argument ***var*** contains a value to be displayed. For an instance, the program code ***"putChar(char c)"*** displays the value of c on monitor screen, where c is of character type. Similarly, ***putFloat(float x)*** will display value of x on monitor screen with two digits after the decimal point.

Using any of the above functions, write a C functions ***print(const char *str,...)*** that will accept a ***format*** string and a number of variables, and then will display the contents of the variables on monitor screen.

Some sample input and output are given below–

Sample input	Sample output
print(%d", A);	12
print("%c %f %d", ch, x, A);	P 12.34 40
print("%d %d %f %c", A, B, X, ch);	23 11 45.23 D

(b) Write down a preprocessor named ***pyramid(n)*** which will print the number pyramid with n rows.

For n = 3	For n = 4	For n = 5
1 212 32123	1 212 32123 4321234	1 212 32123 4321234 543212345

Why is a preprocessor faster than a function call?

(c) Write a C program that will display the details of program running configuration environments on the monitor screen.

3. (a) Consider the following declaration:

static float table $[2][3][2] = \{\{\{1.1, 1.2\}, \{1.3, 1.4\}, \{1.5, 1.6\}\}, \{\{2.1, 2.2\}, \{2.3, 2.4\}, \{2.5, 2.6\}\}\}$

The floating values are stored inside the memory addresses starting from 100 to 116 as given below (assume 2 bytes for floating points)

0^{th} 2D Array						1^{st} 2D Array					
1.1	1.2	1.3	1.4	1.5	1.6	2.1	2.2	2.3	2.4	2.5	2.6
100	102	104	106	108	10A	10C	10E	110	112	114	116

Contd P/4

Contd... Q. No. 3(a)

Fill up the blank in the following table under title of "values of the given part".

Part of Command	Values of the given part
*(table+1)	
((table +1) +1)	
((table + 1))	
***(table +1)	
((table + 1) + 1)	

(b) (i) Write an appropriate declaration for the following situation:

"Declare a pointer to a function that accepts an argument which is an array of pointers to integer quantities and returns a pointer to a character."

(ii) Explain the purpose of the following declaration.

float (*x[20])(int *a);

(c) Write a C program that will accept a file name as the command line argument and print the contents of the file to a printer. If the file name is not provided through the command line, then the program will ask for the file name, read the file name from the standard input device and then print the contents of the file to a printer.

4. (a) A binary search tree, abbreviated to BST, is a binary tree where all the element in the left subtree of a node are less than or equal to the element at the node and all the element at the right subtree are higher than it. A BST can be stored in a 1D array where the root of the tree is stored at index 1. node(*i*) represent the element stored in the index *i*. Left child of node(i) is node (2*i) and the right child of node(i) is node(2*i+1). The value of an element -1 in an index means there is no element at that index.

(i) Write a recursive C function that will insert a node in the given tree. If the tree does not exist, inserting a node means that the node become the root of tree. Use the following function prototype.

void BST_insert(int node, int x); //element x is to be inserted into the BST

(ii) The height of a node of the tree is defined as follows: the height of the root is zero. If a node is at a height of n, then its left and right child are at the height of $n + 1$. Write a recursive C function that will return the height of a given node.

(b) Write down the program for sorting a list of integers using the Merge Sort algorithm.

Contd P/5

SECTION – B

There are **FOUR** questions in this section. Answer any **THREE**.

Please read carefully, some questions might have additional restrictions such as not allowing the use of any library functions except the I/O related ones.

5. (a) Assume **int x** has been declared and initialized to an unknown value. For each of the following, indicate if the expression always evaluates to true, or if it could sometimes be false. In the latter case, indicate a counter example by providing a value for x that falsifies the claim:

   (i) $(x != 0) || (x == 0)$

   (ii) $(x > 0) || (x < 0)$

   (iii) $(((x >> 1) << 1) | (x \& 1)) == x$

   (iv) $(x \wedge (\sim x)) == 0$

   (b) The ceiling of a fractional number is *the smallest integer greater than or equal to it.* For example, the ceiling of 3.12 is 4, whereas the ceiling of -3.12 is -3. Write a C program that will take a fractional number (float) as input and prints its ceiling. **You are not allowed to use any math library function.**

   (c) Write down a program that will determine the summation of the following series:

$$\log_2 2 \_ \log_2 4 + \log_2 8 + \cdots\cdots\cdots + \log_2 n$$

   Assume that $n = 2^m$ for some integer $m > 0$ and **m** will be input to your program. **You are not allowed to use any library function except the I/O related ones.**

   (d) *Twin primes* are a pair of prime numbers that differ by 2. For example, 3 and 5 are twin primes. Some more example of twin primes are (5, 7), (11, 13) etc. Write down a program to find all twin primes less than a positive number $n(>1)$ entered by the user. Display the output as follows:

$$(3, 5)$$
$$(5, 7)$$
$$\ldots$$

6. (a) What is the output of the following code segment (write NO OUTPUT if you think there is an error in the program segment)?

```
int i = 4, j = 5, k;
k = i+++j;
printf("%d %d %d, i, j, k);
```

Contd P/6

Contd... Q. No. 6

(b) Given two integers n and x as parameter where $n > x$ and $x > 1$, write down a C function

int powerOfX (int n, int x);

that will check whether $n = x^y$, for some $y >= 0$, i.e. the number n is a power of x or not and return the value of y if so. Suppose n = 125 and x = 5 then the function will return 3 because $125 = 5^3$. The function will return -1 if n cannot be expressed as a power of x.

(c) In mathematics, a **Fermat number** (named after the great mathematician *Pierre de Fermat* who first studied them) is a positive integer of the form.

$$F_t = 2^{(2^t)} + 1$$

where t is a nonnegative integer. The first few Fermat numbers are:

3, 5, 17, 257, 65537, 4294967297, 18446744073709551617, ...

Write down a program that will take an integer y as input and will determine whether the number is a Fermat number or not using the function int powerOfX (int n, int x) that you have developed in question 6(b).

(d) Write down a program to calculate $\sqrt{x}$ where is a positive double value.

7. (a) Misir Ali is a new C programmer and wrote the following code to determine the largest of three integers *a, b,* and *c*:

```
if (a > b > c) max = a;
else if (b > a > c) max = b;
else max = c;
```

Do you think he is a good programmer? What will be the value of *max* if a = 20, b = 10, and c = 5. Write a **one line statement** using conditional operator (?:) to find the maximum value of three numbers a, b, c. You may use as many conditional operators as needed but it has to be a single line statement.

(b) Complete the definition of the C function maxCount whose prototype is shown below. The function should return the number of times that the global maximum appears in the array list. The global maximum is defined to be the number in the array with the largest value. For example, in the list 3, 9, 7, 5, 9, 8, 2, 4, 9, the global maximum is 9 and it appears 3 times. The maxCount function returns 3 in this case. Assume that list has at least one element. The number of elements in the array is given in the variable length.

int maxCount (int list[], int length);

Contd P/7

Contd... Q. No. 7

(c) Write down a function **nonrepeated** whose prototype is shown below. The function will return the number of **non-repeated** elements in a **sorted** array *a*. The length of the array is *n*. Assume that the array has at least one element and the elements are sorted in the ascending order.

```
int nonrepeated (int a[], int n);
```

8. (a) The following code was written to interchange the value of a and b;

```
a = a^ b;
b = a^ b;
a = a^ b;
```

Do you think the above code will work if *a* and *b* contain the same value? Why or why not, show with an example.

(b) Given a set of numbers where all elements occur even number of times except one number, find the odd occurring number using XOR operator (^). Assume that all numbers are integers and stored in an array of length n (where n is odd). Ask the user to enter the value of n and all numbers at the beginning of your program. For instance, if the set is $\{34, 52, 33, 52, 78, 78, 34, 33, 33\}$, the output will be 33.

(c) Consider the following two definitions of the same macro:

```
#define square_1(x)  x * x
#define square_2(x)  (x) * (x)
```

Give an example here these two macros behave differently, and explain why square_2 is probably a better definition?

(d) Consider the following macro:

```
#define max(a, b) ((a) > (b) ?(a) : (b))
```

is there any problem when we execute the following statement using the above macro?

```
int z = max(i++, j++);
```

Redefine the macro so that the above problem doesn't exit anymore. Assume that the macro will be applied only to integers.
