B.Tech.

FIRST ODD SEMESTER EXAMINATION, 2009-10

COMPUTER CONCEPT IN PROGRAMMING IN C

(ECS-101)

Time : *3 Hours]* *[Total Marks : 100*

Note : (1) This question paper consists of three sections. Section A contains objective type questions and is of 20 marks. Section B consists of short answer type questions which is of 30 marks and Section C contains long answer type questions of total 50 Marks.

(2) Your answers for Section B and C should be precise and to the point.

(3) Answer to the questions of each section must be done at one place in your answer books.

(4) You are required to attempt all the questions.

SECTION-A

Q. 1. There are total 20 multiple choice questions : $10 \times 1 = 10$

Only one of the answer out of given four choices is correct. Write the correct answer.

(i) In evaluation an expression a + b * C, wbich one of the following is correct

(a) + has higher precedence over *

(b) * has higher precedence over +

(c) both * and + have the same precedence

(d) The order of evaluation does not matter

(ii) A stack is

(a) LIFO (Last in First out)

(b) FIFO (First in First out)

(c) LILO (Last Last out)

(d) None of the above

(iii) An array can store

(a) Finite data of similar type

(b) Infinite data of similar type

(c) Finite data of mix type

(d) All of the above

(iv) How many bytes of storage an unsigned short integer in C language would require

(a) 2 (b) 4 (c) 6 (d) 8

(v) Decimal number 10 can be represented in unary (a number system with base 1) as

(a) 1010 (b) 64

(c) A (d) None of the above

(vi) If k = 5 then tbe value of variable x after the execution of a C statement x = k + + will be

(a) 5

(b) 6

(c) randomly any one of the above

(d) value of x will not depend on k

(vii) Typically an operating system

(a) manages all the hardware resources of the computer

(b) complies a high-level program

(c) Both (a) and (b)

(d) None of the above

(viii) For a C program code for (i = 0; i ≤ 10; i + +) {A}; A will run

(a) 10 times (b) 11 times

(c) 12 times (d) None of the above

(ix) Which of the following is not a functional programming language

(a) SML (b) HASKELL

(c) C (d) LISP

(x) A pointer in C language

(a) is a address of some location

(b) is useful in describing linked list

(c) can he used to access the elements of an array

(d) all of the above.

Ans.1 (i) ->b [Ans] * has higher precedence over +

(ii) ->a [Ans] LIFO (Last in First Out)

(iii) ->a [Ans] Finite data of similar types

(iv) ->a [Ans] 2

(v) ->d [Ans] None of the above

(vi) ->a [Ans] 5

(vii) ->a [Ans] Manages all the hardware resources of the computer

(viii) ->b [Ans] 11 times

(ix) ->c [Ans] C

(x) ->d [Ans] All of the above

Q. 2. State whether the following statements are True of False : $5 \times 1 = 5$

(i) Normal binary operators like + and - can be combined with assignment operator = to form new operators in C Language.

(ii) A compiler translates a High-level program into a machine understandable language.

(iii) An algoritbm might never terminate.

(iv) In C language pointers can be used as a function argument.

(v) MS-WORD may be classified as an application software.

Ans. (i)-> FALSE

(ii)-> TRUE

(iii) -> FALSE

(iv) -> TRUE

(v) -> TRUE

Q. 3. Fill in the blanks : $5 \times 1 = 5$

(i) ..... is used to open a file.

(ii) ..... is used as a statement terminator in C.

(iii) The operator && is an example for ..... operator

(iv) A function is called ..... when it calls itself.

(v) An Editor can be classified as ..... software.

Ans. (i) -> fopen [Ans]

(ii) -> semicolon [Ans]

(iii) -> Logical Operator

(iv) -> recursive [Ans]

(v) -> Application [Ans]

SECTION -B

Q. 4. There are total six questions in this section.

Attempt all questions :

$6 \times 5 = 30$

(a) (i) Write a C program to swap two integer variables without using third variable.

(ii) What is the difference between initialization and asignment of a variable.

Ans. (a) (i) */

```
# include<stdio.h>
# include<conio.h>
void main()
{
  int a, b,
  printf ("\n enter the values of a & b");
scanf ("%d%d", &a, &b);
a = a + b;
b = a - b;
a = a - b;
printf (" the swapped values are a =%d,
b = %d");
  getch( );
}
```

Ans. (ii) Difference 1;

You CANNOT assign to a const variable, whereas you CAN initialize it.

For example ;

const int d = 3; //OK! Initialization

d = 3; //WRONG! cannot assign to const

Difference 2;

Initialization can be done at the time of declaration where as assignment can be done anywhere.

Q. 4. (b) (i) Differentiate between WHILE ... Do and Do... WHILE loops.

(ii) Write a recursive C program to calculate tbe factorial of a given integer.

Ans. (i) The difference between "do while" and "do until" is that a "do while" loops while the test case is true, whereas "do until" loops UNTIL the test case is true (which is equivalent to looping while the test case is false).

The difference between a "do ... while" loop and a "while ()" loop is also that the while loop tests its condition before execution of the contents of the loop begins ; the "do" loop tests its condition after it's been executed at least once. As noted above, if the test condition is false as the while loop is entered the block of code is never executed. Since the condition is tested at the bottom of a do loop, its block of code is always executed at least once.

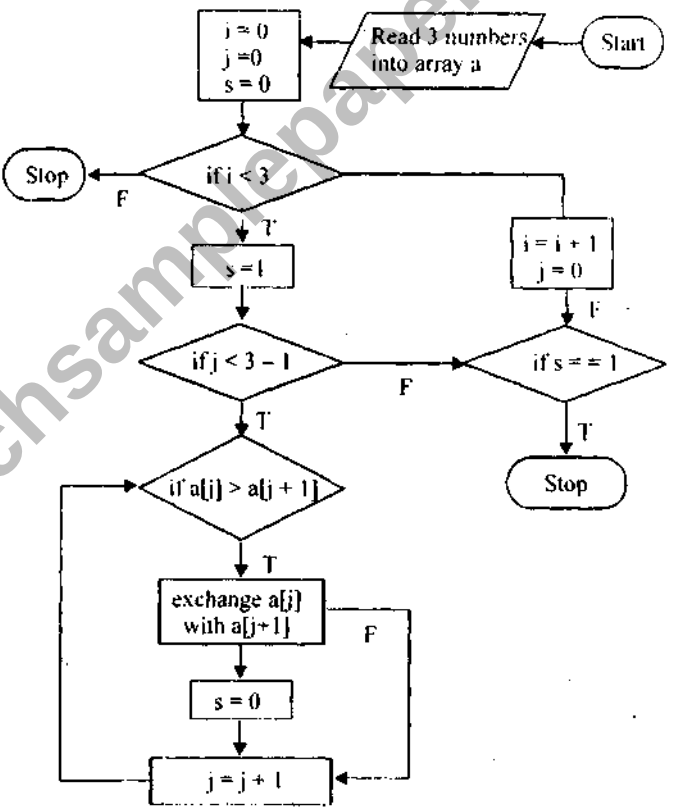
Ans. (b) (ii) */

```
# include<stdio.h>
# include<conio.h>
int factorial(int);
void main ( )
{
  int num, fact;
  printf("\n enter num");
  scanf("%d", & num);
  fact= factorial(num);
  printf(" the factorial of a number is =%d", fact);
  getch( );
}
int factorial(int n)
{
  if (n == 1 )
{
  return(1);
}
else
{
  return(n*factorial(n - 1));
}
```

Q. 4. (c) (i) Draw a flow chart to sort tnree integers.

(ii) What is dynamic memory allocation ? Explain malloc function.

Ans. (i) using bubble sort to sort 3 numbers entered in an array a



(ii) In computer science, dynamic memory allocation (also known as heap-based memory allocation) is the allocation of memory storage for use in a computer program during the runtime of that program. Dynamically allocated memory exists until it is released either explicitly by the programmer, or by the garbage collector.

malloc is a subroutine for performing dynamic memory allocation in the C and C + + programming languages. It is part of the standard library for both languages.

The malloc function is one of the functions is standard C to allocate memory.

Its function prototype is

void*malloc(size_t size);

which allocates size bytes of memory.If the allocation succeeds, a pointer to the block of memory is returned, otherwise a NULL pointer is returned.

Memory allocated via malloc will continue to exist until the program terminates or the memory is explicitly deallocated by the programmer (that is, the block is said to be "freed"). This is achieved by use of the free function.

Q. 4. (d) (i) Write a C program to sequentially search a given integer element from a given list of numbers.

(ii) What is the purpose of using Structures in C ? Explain with the help of a suitable example.

Ans. (i)

```
# include<stdio.h>
# include<conio.h>
# define MAX 10
void main ( )
{
int num, list [MAX],i;
clrscr( );
printf("enter the number to be searched");
scanf("%d", & num);
printf ("\enter the elements of the list");
for(i=0;i<10; i+ +);
{
  scanf("%d", &list[i]);
}
i=0;
while(i<MAX && num!=list[i])
{
  i+ +;
}
if(i<MAX)
  printf("\nthe number is found at=%d", i+1);
else
  printf("\nthe number is not found");
getch( );
}
```

(ii) Arrays are used to store large set of data and manipulate them but the disadvantage is that all the elements stored in an array are to be of the same data type. If we need to use a collection of different data type items it is not possible using an array. When we require using a collection of different data items of different data types we can use a structure. Structure is a method of packing data of different types. A structure is a convenient method of handling a group or related data items of different data types.

structure definition;

general format;

```
struct tag_name
{
data type member1;
data type member 2;
...
...
}
```

Example :

```
struct lib_books
{
char title[20];
char author[15];
int pages;
float price;
};
```

the keyword struct declares a structure to holds the details of four namely title, author pages and price. These are members of the structures. Each member may belong to different or same data type. The tag name can be used to define objects that have the tag names structure. The structure we just declared is not a variable by itself but a template for the structure.

We can declare structure variables using the tag name any where in the program. For example the statement,

struct lib_books book1,book2,book3;

declares book1,book2, book3 as variables of type struct lib_books each declaration has

four elements of the structure lib_books. Structures do not occupy any memory until it is associated with the structure variable such as book1. The template is terminated with a semicolon.

Q. 4. (e) (i) Find the value of X in the equation $(1230)_4 = X_6$.

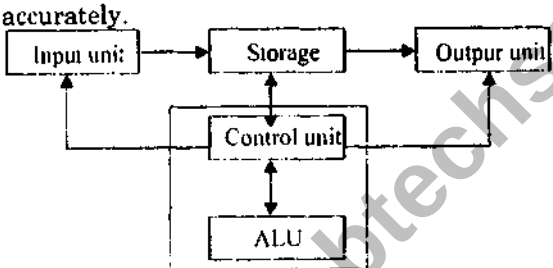
(ii) Draw the functional block diagram of a digital Computer and discuss its components in brief.

Ans. (i) (1230) base 4 = X base 6

1. (1230) base 4 = (1*4^3) + (2 * 4^2) + (3 * 4^1) + (0 * 4^0)
   = (108) base 10
2. Now converting base 10 value to base 6 Gives 300 base 6 because (3 * 6^2) + (0 * 6^1) + (0 * 6^0)

So, X is 300

(ii) A computer can process data, pictures, sound and graphics. They can solve highly complicated problems quickly and accurately.



Input Unit : Computers need to receive data and instruction in order to solve any problem .Therefore we need to input the data and instructions into the computers. The input unit consists of one or more input devices. Keyboard is the one of the most commonly used input device. Other commonly used input devices are the mouse, floppy disk drive, magnetic tape, etc. All the input devices perform the following functions.

- Accept the data and instructions from the outside world.
- Convert it to a form that the computer can understand.
- Supply the converted data to the computer system for further processing.

Storage Unit : The storage unit of the computer holds data and instructions that are entered through the input until, before they are processed. It preserves the intermediate and final results before these are sent to the output devices. It also saves the data for the later use. The various storage devices of a computer system are divided into two categories.

1. Primary Storage : Stores and provides vary fast. This memory is generally used to hold the program being currently executed in the computer, the data being received from the input unit, the intermediate and final results of the program. The primary memory is temporary in nature. The data is lost, when the computer is switched off. In order to store the data permanently, the data has to be transferred to the secondary memory.

The cost of the primary storage is more compared to the secondary storage. Therefore most computers have limited primary storage capacity.

2. Secondary Storage : Secondary storage is used like an archive. It stores several programs, documents, data bases etc. The programs that you run on the computer are first transferred to the primary memory before it is actually run. Whenever the results are saved, again they get stored in the secondary memory. The secondary memory is slower and cheaper than the primary memory. Some of the commonly used secondary memory devices are Hard disk, CD, etc.,

Memory Size : All digital computers use the binary system, i.e. 0's. Each character or a number is represented by an 8 bit code.

The set of 8 bits is called a byte.

A character occupies 1 byte space.

A numeric occupies 2 byte space.

Byte is the space occupied in the memory.

The size of the primary storage is specified in KB (Kilobytes) or MB (Megabyte). One KB is equal to 1024 bytes and one MB is equal to 1000KB.The size of the primary storage in a typical PC usually starts at 16MB. PCs having 32 MB, 48 MB, 128 MB, 256 MB memory are quite common.

Output Unit : The output unit of a computer provides the information and results of a computation to outside world. Printers. Visual Display Unit (VDU) are the commonly used output devices. Other commonly used output devices are floppy disk drive, hard disk drive, and magnetic tape drive.

Arithmetic Logical Unit : All calculations are performed in the Arithmetic Logic Unit (ALU) of the computer. It also does comparison and takes decision. The ALU can perform basic operations such as addition, subtraction, multiplication, division, etc and does logic operations viz, >, <, =, 'etc.Whenever calculations are required, the control unit transfers the data from storage unit to ALU once the computations are done, the results are transferred to the storage unit by the control unit and then it is send to the output unit for displaying results.

Control Unit : It controls all other units in the computer. The control unit instructs the input unit, where to store the data after receiving it from the user. It controls the flow of data and instructions from the storage unit to ALU. It also controls the flow of results from the ALU to the storage unit. The control unit is generally referred as the control nervous system of the computer thet control and synchronizes its working.

Central Processing Unit :

The control unit and ALU of the computer are together known as the Central Processing Unit (CPU).

The CPU is like brain performs the following functions :

- It performs all calculations.
- It takes all decisions.
- It controls all units of the computer.

SECTION-C

Q. 5. This section contains SEVEN programming questions. Attempt any FIVE questions. All answers must contain Flow chart /Algorithm for your program ogic : **10 × 5 = 50**

(a) Write a C program to read in 10 integer numbers and print their average, minimum and maximum numbers.

Ans. (a) */

```
#include <stdio.h>
void main( )
{
  int a[10],i;
float average= 0;
int min,max;
for (i=0;i<10;i+ +)
{
  printf("\nEnter value");
  scanf("%d",&a[i]);
  average+ =a[i];
}
average = average / 10;
prinf("\nAverage is : %f"average);
min=max=a[0];
for(i=1; i<10;+ + i)
  if(min>a[i])
                    min=a[i];
  if(max<a[i])
                    max=a[i];
}
prinf("\nMin, Max is : %d %d", min,
max);
}
```

Q. 5. (b) Write a C program to add, multiply two N × N matrix.

Ans. */

```
#include<stdio.h>
#define MAX 10
void main( )
{
      int a[MAX][MAX],b[MAX][MAX],
            c[MAX][MAX];
      int i,j,k,N;
      print ("Enter the value of N for NxN
      Matrix <= 10 :");
      scanf("%d, &N);
      for(i=0;i<N;+ +i)
      {
        printf("\n Matrix1 :: Enter the
        Element %d of row",i+1 );
        printf("\n Matrix2 :: Enter the
Element %d of row",i+1 );
        for(j=0; j<N;+ +j)
```

```
        {
            scanf("%d", &a[i][j]);
            scanf("%d", &b[i][j];
            c[i][j] = a[i][j] + b[i][j];
        }
    }
    for(i=0;i<N;+ + i)
    {
      printf("\n");
      for(j=0;j<N;+ + j)
      {
            printf("%d\t",c[i][j]);
      }
    }
      for(i=0;i<N;+ +i)
    {
    printf("\n");
    for(j=0;j<N;+ + j)
    {
      c[i][j] = 0;
      for(k=0;k<N;+ + k)
      {
      c[i][j]+ = a[i][k] * b[k][j];
      }
    }
    for(i=0;i<N;+ + i)
    {
      printf("\n");
      for(j=0;j<N;+ + j)
      {
            printf("%d\t",c[i][j]);
      }
    }
}
}
```

Q. 5. (c) Write a simple database program in C which stores personal details of 100 persons such as Name, Date of Brith, Address, Phone number etc.

Ans. */

```
#include<stdio.h>
struct database
{
        char name[25];
        char dob[8];
        char address[40];
        char phone[15];
        };
        main( )
        {
        struct database data[100];
        int i=0;
        printf("Enter the data of 100
        people");
          for(i=0;i<100;+ + i)
        {
              printf("Enter data for
              %d person",i+1
              scanf("%s",data[i].name);
              scanf("%s",data[i].dob);
              scanf("%s",data[i].address);
              scanf("%s",data[i].phone);
        }

        for (i=0;i<100;+ + i)
        {
            printf("%s",data[i].name);
              printf("%s",data[i].dob);
              printf("%s",data[i].address);
              printf("%s",data[i].phone);
        }
        }
```

Q. 5. (d) Write a C program which reverses the digits of the integer input given to it. For example an input 65367 is outputted as 76356.

Ans. */

```
#include <stdio.h>
#include <math.h>
void main()
{
  int num,count=0,temp;
  scanf("%d", &num);
  temp=num;
  while(temp!=0)
  {
      temp=temp/10
      + + count;
  }
  temp= 0;
  while(count!=0)
  {
       temp = temp + (num % 10)*
                        pow(10,count-1);
       --count;
       num = (num>=10)?(num/10):num;
  }
```

```
    printf("%d",temp);
}
```

Q. 5. (e) Write a C program to calculate the sum of the following series upto 50 terms

SUM = - $1^2 + 3^3 - 5^3 + 7^3 - 9^3 + 11^3 - ..$

Ans. */

```
#include <stdio.h>
#include <math.h>
void main()
{
  int i= 1;
  int sum = 0 ;
  int count =0;
  for(count=0; count<50;+ +count)
  {
      if(count%2==0)
      {
      sum + = pow(-i,3);
      }
      else
      {
          sum + = pow(-i,3);
      }
      i+ = 2;
  }
  printf("SUM is %d",sum);
}
```

Q. 5. (f) Write a C program to print n^{th} Fibonacci number.

Ans. */

```
#include <stdio.h>
void main()
{
  int n;
  int a=0,b=1,c=0;
  printf("Enter the Nth number required of Fibonacci Series");
  scanf ("%d", &n);
  while(n!=0)
  {
      c=a+b;
      a=b;
      b=c;
      --n;
  }
      printf("Result is %d",c);
}
```

Q.5. (g) Write a C program to arrange given n strings in lexicographical order.

Ans.*/

```
#include <stdio.h>
#include <string.h>
void main( )
{
      int i,j=0;
      char*strings[8]={
                        "abc",
                        "abcd",
                        "ab",
                        "dfg",
                        "sdss",
                        "wewqqq",
                        "abcde",
                        "fgds"
                        };
char*temp;
      for(i=0;i<10;i+ +)
      for(j=1 ;j<10-i;j+ +)
      {
          if(strcmp(strings[j],str
      ings[j-1])<0)
          {
                        temp = strings[j];
                  strings[j]=strings[j-1];
                      strings[j-1]=temp;
      }
}
for(i=0;i<8;+ + i)
{
          printf("%s",strings[i]);
}
}
```