

Course Code : **MCS-011**
Course Title : **Problem Solving and Programming**

Q1: (a) Write a simple program to find the size of different basic data types in C.

Hint: Program to find the size of different basic data types in C

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x;
float y;
char z;
double d;
long l;
clrscr();
printf("size of integer= %d",sizeof(x));
printf("\nsize of float= %d",sizeof(y));
printf("\nsize of char= %d",sizeof(z));
printf("\nsize of double= %d",sizeof(d));
printf("\nsize of long= %d",sizeof(l));
getch();
}
```

(b) Write a program in C for arithmetic operations between two integers. Your program should guide users with proper message/menu on the console.

Hint: program in c for arithmetic operations between two integers. your program should guide users with proper message/menu on the console.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num1,num2,sum,difference,mul,choice;
float div;
start :
clrscr();
printf("\n1.addition");
printf("\n2.subtraction");
printf("\n3.multiplication");
printf("\n4.division");
printf("\n5.exit.");
printf("\n\nenter your choice:");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("\nenter first number");
scanf("%d",&num1);
printf("\nenter second number");
scanf("%d",&num2);
printf("\n sum of two numbers=%d",(num1+num2));
break;
case 2:
printf("\nenter first number");
scanf("%d",&num1);
```

```

printf("\nenter second number");
scanf("%d",&num2);
printf("\n difference of two numbers=%d",(num1-num2));
break;
case 3:
printf("\nenter first number");
scanf("%d",&num1);
printf("\nenter second number");
scanf("%d",&num2);
printf("\n product of two numbers=%d",(num1*num2));
break;
case 4:
printf("\nenter first number");
scanf("%d",&num1);
printf("\nenter second number");
scanf("%d",&num2);
printf("\n division of two numbers=%d",(num1/num2));
break;
case 5:
printf("\npress enter to exit");
getch();
exit(0);
default :
printf("\nyou have entered wrong choice:\n please enter new choice:");
goto start;
}
}

```

(c) Write a function Pali(Sting S) to find whether S is palindrome or not.

Hint: function pali (**sting s**) to find whether s is palindrome or not.

```

pali (char s[])
{
int length,i,j,flag=0;
length=strlen(s);
for(i=0,j=length-1;i<(length/2),j>(length/2);i++,j--)
{
if(s[i]!=s[j])
flag=1;
}
if(flag==1)
{
printf("\nstring is not pallendrome");
}
else
{
printf("\nstring is pallendrome");
}
getch();
}

```

Q2: (a) Write a C program to print this triangle:

```
*
***
*****
*****
*****
*****
*****
```

Hint:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int i,j;
    for(i=0;i<12;i+=2)
    {
        for(j=i;j>=0;j--)
        {
            printf("*");
        }
        printf("\n");
    }
    getch();
}
```

(b) Write a C program to compute the average marks in a subject of all the students in a class.

Hint: C program to compute the average marks in a subject of all the students in a class.

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int marks, i, noofstudents, sum=0;
    float avg;
    clrscr();
    printf("enter no of students");
    scanf("%d",&noofstudents);
    printf("\n enter marks of students");
    for(i=0;i<noofstudents;i++);
    {
        scanf("%d",&marks);
        sum+=marks;
    }
    printf("\naverage of class=%f",(sum/noofstudents));
    getch();
}
```

Q3 (a) Write a program to get the value of sin (x) using a library function , when x is given in degrees.

Hint: #include<stdio.h> #include<conio.h>

#include<math.h>

```
void main() { float x,value; printf("\n Enter a number to get its Sin(x) value="); scanf("%f",&x); clrscr();
value=sin(x); printf("\n Value of Sin(x)=%f",value); getch(); }
```

(b) There are 20 integer values stored in an array. Write a programme to find largest and smallest value store.

Hint: There are 20 integer values stored in an array. Write a programme to find largest and smallest value store.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num[20],largest,smallest,i;
clrscr();
printf("\nenter twenty numbers");
for(i=0;i<20;i++)
{
scanf("%d",&num[i])
}
largest=num[0];
smallest=num[0];
for(i=1;i<20;i++)
{
if(num[i]>largest)
largest=num[i];
if(num[i]<smallest)
smallest=num[i];
}
printf("\nsmallest of 20 numbers=%d",smallest);
printf("\nlargest of 20 numbers=%d",largest);
getch();
}
```

(c)Write a c program for binary addition of two 8 bit numbers.

Hint:

```
# include<stdio.h>
# include<conio.h>
# include<stdlib.h>
void binaryadd(int,int*,int*,int); /* funtion to add to binay numbers */
int a[8],b[8],c[8]; /* as we want 8 bits , here extern data type is required */
void main()
{ clrscr();
int n1,n2,i=0,carry=0,*m,*n,k,x,y,d[8]={ 1,0,0,0,0,0,0,0};
char op;
gotoxy(1,6);printf("Enter 1st Numbers :");
scanf("%d",&n1);
gotoxy(1,7);printf("Enter 2nd Numbers :");
scanf("%d",&n2);
oper: /* label to return back if user gives wrong operator */
gotoxy(1,11);printf(" ");
gotoxy(1,8);printf("\nEnter the operator(+,-) :");
op=getch();
x=n1; /* for future refence in case of subtraction */
y=n2;
if(op == '+' || op == '-')
{ while(n1!=0) /* converting 1st number to its binary equivalent */
{ a[i]=n1 % 2;
n1/=2;
i++;
```

```

}
/* printing 1st number */
printf("\nThe binary of %d is : ",x);
for(k=7;k>=0;k--)
printf("%d",a[k]);
printf("\n");
i=0;
while(n2!=0)
{ b[i]=n2 % 2; /* converting 2nd number to its binary equivalent */
n2/=2;
i++;
}
/* printing binary of 2nd number */
printf("\nThe binary of %d number is :",y);
for(k=7;k>=0;k--)
printf("%d",b[k]);
printf("\n");
}
switch(op)
{ case '+':
i=0;
m=a;
n=b;
carry=0;
binaryadd(i,m,n,carry); /*function called for binary add */
printf("\nThe addition is :");
for(i=7;i>=0;i--)
{ printf("%d",c[i]);
}
break;
case '-':
for(i=0;i<=7;i++)
{ if(b[i]==0)
b[i]=1; /* 1's complement */
else
b[i]=0;
}
printf("\nThe 1's complement of %d is : ",y);
for(i=7;i>=0;i--)
printf("%d",b[i]);
i=0;
m=b;
n=d;
carry=0;
binaryadd(i,m,n,carry); /* called for 2's complement */
printf("\nThe 2's complement of %d is : ",y);
for(i=7;i>=0;i--)
printf("%d",c[i]);
for(i=0;i<=7;i++)
b[i]=c[i];
i=0;
m=a;
n=b;
carry=0;
binaryadd(i,m,n,carry);
if(x>y)

```

```

{ printf("\nThe subtraction is : ");
for(i=7;i>=0;i--)
printf("%d",c[i]);
if(x < y)
{ for(i=7;i>=0;i--)
{ if(c[i]==0)
c[i]=1;
else
c[i]=0;
}
i=0;
m=c;
n=d;
carry=0;
binaryadd(i,m,n,carry);/* again 2's complement for negative answers */
printf("\nThe subtraction is : ");
printf("-");
for(i=7;i>=0;i--)
printf("%d",c[i]);
}
exit(0);
break;
default:
gotoxy(1,11);printf("WRONG OPTION");
getch();
goto oper;
}
getch();
}
void binaryadd(int i,int *m,int *n,int carry)
{ if(*m==1 && *n==1 && carry==1)
{ c[i]=1;
carry=1;
}
if(*m==1 && *n==1 && carry==0)
{ c[i]=0;
carry=1;
}
if(*m==0 && *n==1 && carry==0)
{ c[i]=1;
carry=0;
}
if(*m==0 && *n==1 && carry==1)
{ c[i]=0;
carry=1;
}
if(*m==1 && *n==0 && carry==0)
{ c[i]=1;
carry=0;
}
if(*m==1 && *n==0 && carry==1)
{ c[i]=0;
carry=1;
}
if(*m==0 && *n==0 && carry==0)
{ c[i]=0;

```

```

carry=0;
}
if(*m==0 && *n==0 && carry==1)
{ c[i]=1;
carry=0;
}
i++;
m++;
n++;
if(i<=8) /* Base value of recursion */
binaryadd(i,m,n,carry); /* recursion till we reach 8 bit of number */
}

```

Q4: (a) Using Pointers,

i) Write the character-counting function

Hint:

```

int CharCount(char *str) { int count = 0; if (str != NULL) { for (; *(str + count) != '\0'; ++count);
} return count; }

```

ii) Write the string-concatenation program

Hint: string_concatenate(char *s1, char *s2) { int i, j; char s3[50]; for(i=0; i<strlen(s1); i++) s3[i]=s1[i]; for(j=0; j<strlen(s2); j++) s3[i+j]=s2[j]; printf("%s", s3); }

(b) Explain pointer arithmetic with example. Also explain advantages of *malloc* and *calloc*

Hint: A pointer is a variable that contains the memory location of another variable. The syntax is as shown below. You start by specifying the type of data stored in the location identified by the pointer. The asterisk tells the compiler that you are creating a pointer variable. Finally you give the name of the variable.

type * variable name **Example:**

```
int *ptr; float *string;
```

C allows us to add integers to or subtract integers from pointers as well as to subtract one pointer from the other. We can also use short hand operators with the pointers `p1+=`; `sum+=*p2`; etc.,

we can also compare pointers by using relational operators the expressions such as `p1 > p2`, `p1 == p2` and `p1 != p2` are allowed. /*Program to illustrate the pointer expression and pointer arithmetic*/ #include <stdio.h> main() { int ptr1, ptr2; int a, b, x, y, z; a=30; b=6; ptr1=&a; ptr2=&b; x=*ptr1 + *ptr2 - 6; y=6 * - *ptr1 / *ptr2 + 30; printf("\nAddress of a %u", ptr1); printf("\nAddress of b %u", ptr2); printf("\na=%d, b=%d", a, b); printf("\nx=%d, y=%d", x, y); ptr1=ptr1 + 70; ptr2=ptr2; printf("\na=%d, b=%d", a, b); }.

Advantage of Malloc and Calloc

`malloc()` allocates byte of memory, whereas `calloc()` allocates block of memory.

`Calloc(m, n)` is essentially equivalent to `p = m * malloc(n); memset(p, 0, m * n);` The zero fill is all-bits-zero, and does not therefore guarantee useful null pointer values (see section 5 of this list) or floating-point zero values. Free is properly used to free the memory allocated by `calloc`.

`Malloc(s);` returns a pointer for enough storage for an object of `s` bytes. `Calloc(n, s);` returns a pointer for enough contiguous storage for `n` objects, each of `s` bytes. The storage is all initialized to zeros.

Simply, `malloc` takes a single argument and allocates bytes of memory as per the argument taken during its invocation. Where as `calloc` takes two arguments, they are the number of variables to be created and the capacity of each variable (i.e. the bytes per variable).

Q5: (a) Write a C program for Tower of Hanoi problem with a example of 4 disks

Hint: #include <conio.h>
#include <stdio.h>
void main()
{
void hanoi(char,char,char,int);
char t1='A',t2='B',t3='C';
int n;
clrscr();
printf("\n Enter the no. of disks on Tower A:");
scanf("%d",&n);
if(n<1)
{
printf("\n Nothing to move");
}
else
hanoi(t1,t2,t3,n);
getch();
}
void hanoi(char t1,char t2,char t3,int n)
{
static int step=0;
step++;
printf("\n %c %c %c %d",t1,t2,t3,n);
if(n==1)
{
printf("\n Move top disk from Tower %c ----> %c",t1,t2);
return;
}
hanoi(t1,t3,t2,n-1);
printf("\n %c %c %c %d",t1,t2,t3,n);
printf("\n Move top disk from Tower %c ----> %c",t1,t2);
printf("\n %c %c %c %d",t1,t2,t3,n);
hanoi(t3,t2,t1,n-1);
printf("\n %c %c %c %d steps=%d",t1,t2,t3,n,step);
}
}

(b) Write a C program to store students records in a file.

Hint: #include <stdio.h>
#include<conio.h>
#include<math.h>
const int MAX_CLASS_SIZE = 500;
const int MAX_SURNAME_SIZE = 20;
const int MAX_FNAME_SIZE = 40;
struct Student {
char id[10];
char surname[MAX_SURNAME_SIZE];
char firstname[MAX_FNAME_SIZE];
};
typedef Student StudentGroup[];
void getStudent(Student& s);
void displayStudent(Student s);
void getData(StudentGroup arr, int& howMany);


```

void bubSort(StudentGroup arr, int howMany);
void displayReport(StudentGroup arr, int howMany);
void main()
{
    Student s[MAX_CLASS_SIZE];
    int n = 0;
    getData(s,n);
    bubSort(s,n);
    displayReport(s,n);
}
void getStudent(Student& s)
{
    Scanf("%s", s.id);
    Scanf("%s", s.surname);
    getline(s.firstname, MAX_FNAME_SIZE); // collects rest of input line
}
void displayStudent(Student s)
{
    Printf("%s %s %s", s.id , s.firstname,s.surname);
    // this was screwed up because I had ' ' instead of " "
}
void getData(StudentGroup arr, int& howMany)
{
    howMany = 0;
    while (1)
    {
        getStudent(arr[howMany]);
        if(cin) // successful data read of 1 student
        {
            howMany++;
        }
    }
}
void bubSort(StudentGroup arr, int n)
{
}
void displayReport(StudentGroup arr, int howMany)
{
    for (int i=0; i<howMany; i++)
    {
        displayStudent(arr[i]);
    }
    Printf( "Finshed writing report\n");
}

```