BCA103P: C PROGRAMMING LAB

1) Write a C Program to find the roots of the given quadratic equation using if-else if statement.

```
#include <stdio.h>
#include <conio.h>
#include<math.h>
void main()
{
   int a,b,c,d;
   float x1,x2;
   clrscr();

   printf("enter the value of a,b & c\n");
   scanf("%d%d%d",&a,&b,&c);
   d=b*b-4*a*c;
   if(d==0)
   {
     printf("Both roots are equal\n");
     x1=-b/(2.0*a);
     x2=x1;
     printf("First  Root x1= %f\n",x1);
     printf("Second Root x2= %f\n",x2);
   }
   else if(d>0)
{
  printf("Both roots are real and diff-2\n");
  x1=(-b+sqrt(d))/(2*a);
  x2=(-b-sqrt(d))/(2*a);
  printf("First  Root x1= %f\n",x1);
  printf("Second Root x2= %f\n",x2);
}
else
   printf("Root are imeginary\n No Solution \n");
   getch();
}
```

2) Write a menu driven C program using switch-case to find: (a) Prime or not (b) Factorial of N(c)Even or odd number.

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
main()
{
   int c,n,b,i,j,g;
   printf("Select one :\n 1. factorial.\n2. prime nmber:\n3. even or odd:\n4. exit.\n Enter choice:");
   scanf("%d",&c);
   switch(c)
   {
      case 1:
         printf("enter no.:");
         scanf("%d",&n);
         b=1;
         for(i=n;i>0;i--)
            b=b*i;
         printf("factorial is %d",b);
         system("pause");
         break;
      case 2 :
         printf("Enter no.:");
         scanf("%d",&j);
         if(j%2==0)
         {
            printf("it is not a prime numbr.\n");
         }
         else
            printf("it is a prime number.\n");
         system("pause");
         break;
      case 3 :
         printf("enter the :");
         scanf("%d",&g);
         if(g%2==0)
            printf("it is an even number.\n");
         else
            printf("not an even number\n");
         system("pause");
         break;
      case 4 :
         exit(0);
   }
   system("pause");
}
```

3) Write a C program to find cos (x) using series cos(x) = 1 – x 2 /2! + x4 /4! - ……xn /n! ]

```
#include <stdio.h>
 int main()
       {

int i, j, n, fact, sign = - 1;

float x, p, sum = 0;

printf("Enter the value of x : ");

 scanf("%f", &x);

printf("Enter the value of n : ");

scanf("%d", &n);

for (i = 2; i <= n; i += 2)

 {

p = 1;

fact = 1;

for (j = 1; j <= i; j++)
{

 p = p * x;

 fact = fact * j;

 }

sum += sign * p / fact; sign = - 1 * sign;

}

printf("cos %0.2f = %f", x, 1+sum);

return 0;

}
```

4) Write a C program to arrange the given set of numbers in ascending and descending order.

```
#include <stdio.h>
 int main() {
      int n, data[100], i, j, temp;

      /* get the number of entries */
      printf("Enter your input for n:");
      scanf("%d", &n);

      /* get the input data */
      for (i = 0; i < n; i++)
            scanf("%d", &data[i]);

      /* sort the given data in ascending order */
      for (i = 0; i < n-1; i++) {
            for (j = i + 1; j < n; j++) {
                  if (data[i] > data[j]) {
                        temp = data[i];
                        data[i] = data[j];
                        data[j] = temp;
                  }
            }
      }

      /* data in ascending order */
      printf("Ascending Order:\n");
      for (i = 0; i < n; i++)
            printf("%d\n", data[i]);

      /* data in descending order */
      printf("\nDescending Order:\n");
      for (i = n-1; i >= 0; i--)
            printf("%d\n", data[i]);

      return 0;
 }
```

5) Write a C program to find product of two N x M matrices.

```
#include <stdio.h>

int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter the number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter the elements of first matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &first[c][d]);

  printf("Enter the number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if (n != p)
    printf("Matrices with entered orders can't be multiplied with each
other.\n");
  else
  {
    printf("Enter the elements of second matrix\n");

    for (c = 0; c < p; c++)
      for (d = 0; d < q; d++)
        scanf("%d", &second[c][d]);

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++) {
        for (k = 0; k < p; k++) {
          sum = sum + first[c][k]*second[k][d];
        }

        multiply[c][d] = sum;
        sum = 0;
      }
    }

    printf("Product of entered matrices:-\n");

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++)
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }

  return 0;
}
```

6) Write a C program to concatenate two strings using pointers.

```
#include<stdio.h>
int main(){
  int i=0,j=0;
  char *str1,*str2,*str3;
  puts("Enter first string");
  gets(str1);
  puts("Enter second string");
  gets(str2);
  printf("Before concatenation the strings are\n");
  puts(str1);
  puts(str2);
  while(*str1){
      str3[i++]=*str1++;
  }
  while(*str2){
      str3[i++]=*str2++;
  }
  str3[i]='\0';
  printf("After concatenation the strings are\n");
  puts(str3);
  return 0;
}
```

7) Write a C program to copy content of one file to another file.

```
#include <stdio.h>
#include <stdlib.h> // For exit()

int main()
{
    FILE *fptr1, *fptr2;
    char filename[100], c;

    printf("Enter the filename to open for reading \n");
    scanf("%s", filename);

    // Open one file for reading
    fptr1 = fopen(filename, "r");
    if (fptr1 == NULL)
    {
        printf("Cannot open file %s \n", filename);
        exit(0);
    }

    printf("Enter the filename to open for writing \n");
    scanf("%s", filename);

    // Open another file for writing
    fptr2 = fopen(filename, "w");
    if (fptr2 == NULL)
    {
        printf("Cannot open file %s \n", filename);
        exit(0);
    }

    // Read contents from file
    c = fgetc(fptr1);
    while (c != EOF)
    {
        fputc(c, fptr2);
        c = fgetc(fptr1);
    }

    printf("\nContents copied to %s", filename);

    fclose(fptr1);
    fclose(fptr2);
    return 0;
      }
```