Introduction to C: Exercises

By G GROUP

Q.1 C program to check odd or even using modulus operator

```
#include<stdio.h>

main()
{
   int n;

   printf("Enter an integer\n");
   scanf("%d",&n);

   if ( n%2 == 0 )
      printf("Even\n");
   else
      printf("Odd\n");

   return 0;
}
```

Q.2 C program to check odd or even using bitwise operator

```
#include<stdio.h>

main()
{
   int n;

   printf("Enter an integer\n");
   scanf("%d",&n);

   if ( n & 1 == 1 )
      printf("Odd\n");
   else
      printf("Even\n");

   return 0;
}
```

Q.3 C program to check odd or even without using bitwise or modulus operator

```
#include<stdio.h>

main()
{
   int n;

   printf("Enter an integer\n");
   scanf("%d",&n);

   if ( (n/2)*2 == n )
```

```
      printf("Even\n");
   else
      printf("Odd\n");

   return 0;
}
```

Q.4 Find odd or even using conditional operator

```
#include<stdio.h>

main()
{
   int n;

   printf("Enter an integer\n");
   scanf("%d",&n);

   n%2 == 0 ? printf("Even number\n") : printf("Odd number\n");

   return 0;
}
```

C programming:

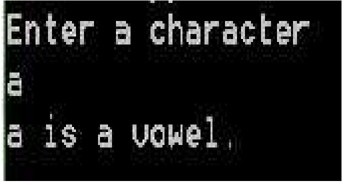
Q.5 c program to check whether input alphabet is a vowel or not

This code checks whether an input alphabet is a vowel or not. Both lower-case and upper-case are checked.

C programming code

```
#include <stdio.h>

main()
{
  char ch;

  printf("Enter a character\n");
  scanf("%c", &ch);

  if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' ||
ch == 'I' || ch =='o' || ch=='O' || ch == 'u' || ch == 'U')
    printf("%c is a vowel.\n", ch);
  else
    printf("%c is not a vowel.\n", ch);

  return 0;
}
```

Output:



```
Enter a character
a
a is a vowel.
```

Q.6 Check vowel using switch statement

```
#include <stdio.h>

main()
{
  char ch;

  printf("Enter a character\n");
  scanf("%c", &ch);

  switch(ch)
  {
    case 'a':
    case 'A':
    case 'e':
    case 'E':
    case 'i':
    case 'I':
    case 'o':
    case 'O':
    case 'u':
    case 'U':
      printf("%c is a vowel.\n", ch);
      break;
    default:
      printf("%c is not a vowel.\n", ch);
  }

  return 0;
}
```

Q.7 Use of Function to check vowel

```
int check_vowel(char a)
{
    if (a >= 'A' && a <= 'Z')
       a = a + 'a' - 'A';   /* Converting to lower case or use a = a +
32 */

    if (a == 'a' || a == 'e' || a == 'i' || a == 'o' || a == 'u')
       return 1;

    return 0;
}
```

C programming:

Q.8 c program to check leap year

c program to check leap year: c code to check leap year, year will be entered by the user.

C programming code

```
#include <stdio.h>

int main()
{
  int year;

  printf("Enter a year to check if it is a leap year\n");
  scanf("%d", &year);

  if ( year%400 == 0)
    printf("%d is a leap year.\n", year);
  else if ( year%100 == 0)
    printf("%d is not a leap year.\n", year);
  else if ( year%4 == 0 )
    printf("%d is a leap year.\n", year);
  else
    printf("%d is not a leap year.\n", year);

  return 0;
}
```

Q.9 add digits of number in c

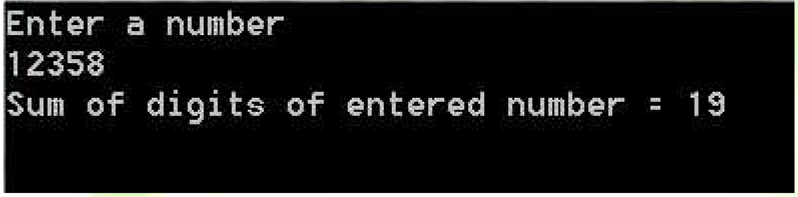
C program to add digits of a number: Here we are using modulus operator(%) to extract individual digits of number and adding them.

C programming code

```
#include <stdio.h>

main()
{
   int n, sum = 0, remainder;

   printf("Enter an integer\n");
   scanf("%d",&n);

   while(n != 0)
   {
      remainder = n % 10;
      sum = sum + remainder;
      n = n / 10;
   }

   printf("Sum of digits of entered number = %d\n",sum);

   return 0;
}
```

For example if the input is 98, sum(variable) is 0 initially
98%10 = 8 (% is modulus operator which gives us remainder when 98 is divided by 10).
sum = sum + remainder
so sum = 8 now.
98/10 = 9 because in c whenever we divide integer by another integer we get an integer.
9%10 = 9
sum = 8(previous value) + 9
sum = 17
9/10 = 0.
So finally n = 0, loop ends we get the required sum.

Output of program:



```
Enter a number
12358
Sum of digits of entered number = 19
```

Q.10 Add digits using recursion

```
#include <stdio.h>

int add_digits(int);

int main() {
  int n, result;

  scanf("%d", &n);

  result = add_digits(n);

  printf("%d\n", result);

  return 0;
}

int add_digits(int n) {
  static int sum = 0;

  if (n == 0) {
    return 0;
  }

  sum = n%10 + add_digits(n/10);

  return sum;
}
```

C programming:

Q.11 Factorial program in c

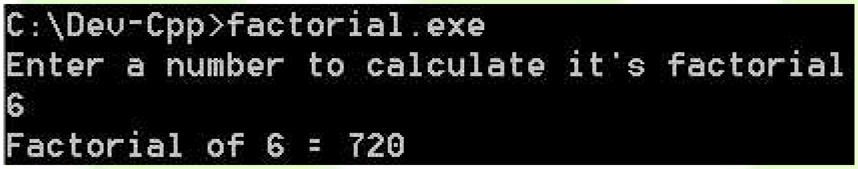
Factorial program in c: c code to find and print factorial of a number, three methods are given, first one uses a for loop, second uses a function to find factorial and third using recursion. Factorial is represented using !, so five factorial will be written as 5!, n factorial as n!. Also
n! = n*(n-1)*(n-2)*(n-3)...3.2.1 and zero factorial is defined as one i.e. 0!=1.

Factorial program in c using for loop

: Here we find factorial using for loop.

```
#include <stdio.h>

int main()
{
  int c, n, fact = 1;

  printf("Enter a number to calculate it's factorial\n");
  scanf("%d", &n);

  for (c = 1; c <= n; c++)
    fact = fact * c;

  printf("Factorial of %d = %d\n", n, fact);

  return 0;
}
```

Factorial
Output of code:



```
C:\Dev-Cpp>factorial.exe
Enter a number to calculate it's factorial
6
Factorial of 6 = 720
```

Q.12 Factorial program in c using function

```
#include <stdio.h>

long factorial(int);

int main()
{
  int number;
  long fact = 1;

  printf("Enter a number to calculate it's factorial\n");
  scanf("%d", &number);

  printf("%d! = %ld\n", number, factorial(number));

  return 0;
}
```

```
long factorial(int n)
{
  int c;
  long result = 1;

  for (c = 1; c <= n; c++)
    result = result * c;

  return result;
}
```

Q.13 Factorial program in c using recursion

```
#include<stdio.h>

long factorial(int);

int main()
{
  int num;
  long f;

  printf("Enter a number to find factorial\n");
  scanf("%d", &num);

  if (num < 0)
    printf("Negative numbers are not allowed.\n");
  else
  {
    f = factorial(num);
    printf("%d! = %ld\n", num, f);
  }

  return 0;
}

long factorial(int n)
{
  if (n == 0)
    return 1;
  else
    return(n * factorial(n-1));
}
```

Q.14 c program to find hcf and lcm

C program to find hcf and lcm: The code below finds highest common factor and least common multiple of two integers. HCF is also known as greatest common divisor(GCD) or greatest common factor(gcf).

C programming code

```
#include <stdio.h>
```

```c
int main() {
  int a, b, x, y, t, gcd, lcm;

  printf("Enter two integers\n");
  scanf("%d%d", &x, &y);

  a = x;
  b = y;

  while (b != 0) {
    t = b;
    b = a % b;
    a = t;
  }

  gcd = a;
  lcm = (x*y)/gcd;

  printf("Greatest common divisor of %d and %d = %d\n", x, y, gcd);
  printf("Least common multiple of %d and %d = %d\n", x, y, lcm);

  return 0;
}
```

Q.15 C program to find hcf and lcm using recursion

```c
#include <stdio.h>

long gcd(long, long);

int main() {
  long x, y, hcf, lcm;

  printf("Enter two integers\n");
  scanf("%ld%ld", &x, &y);

  hcf = gcd(x, y);
  lcm = (x*y)/hcf;

  printf("Greatest common divisor of %ld and %ld = %ld\n", x, y, hcf);
  printf("Least common multiple of %ld and %ld = %ld\n", x, y, lcm);

  return 0;
}

long gcd(long a, long b) {
  if (b == 0) {
    return a;
  }
  else {
    return gcd(b, a % b);
  }
}
```

Q.16 C program to find hcf and lcm using function

```
#include <stdio.h>

long gcd(long, long);

int main() {
  long x, y, hcf, lcm;

  printf("Enter two integers\n");
  scanf("%ld%ld", &x, &y);

  hcf = gcd(x, y);
  lcm = (x*y)/hcf;

  printf("Greatest common divisor of %ld and %ld = %ld\n", x, y, hcf);
  printf("Least common multiple of %ld and %ld = %ld\n", x, y, lcm);

  return 0;
}

long gcd(long x, long y) {
  if (x == 0) {
    return y;
  }

  while (y != 0) {
    if (x > y) {
      x = x - y;
    }
    else {
      y = y - x;
    }
  }

  return x;
}
```

Q.17 Decimal to binary conversion

C program to convert decimal to binary: c language code to convert an integer from decimal number system(base-10) to binary number system(base-2). Size of integer is assumed to be 32 bits. We use bitwise operators to perform the desired task. We right shift the original number by 31, 30, 29, ..., 1, 0 bits using a loop and bitwise AND the number obtained with 1(one), if the result is 1 then that bit is 1 otherwise it is 0(zero).

C programming code

```
#include <stdio.h>

int main()
{
  int n, c, k;

  printf("Enter an integer in decimal number system\n");
  scanf("%d", &n);
  printf("%d in binary number system is:\n", n);

  for (c = 31; c >= 0; c--)
  {
    k = n >> c;

    if (k & 1)
      printf("1");
    else
      printf("0");
  }

  printf("\n");

  return 0;
}
```

Above code only prints binary of integer, but we may wish to perform operations on binary so in the code below we are storing the binary in a string. We create a function which returns a pointer to string which is the binary of the number passed as argument to the function.

Q.18 C code to store decimal to binary conversion in a string

```
#include <stdio.h>
#include <stdlib.h>

char *decimal_to_binary(int);

main()
{
   int n, c, k;
   char *pointer;

   printf("Enter an integer in decimal number system\n");
   scanf("%d",&n);

   pointer = decimal_to_binary(n);
   printf("Binary string of %d is: %s\n", n, t);

   free(pointer);

   return 0;
}

char *decimal_to_binary(int n)
{
   int c, d, count;
   char *pointer;

   count = 0;
   pointer = (char*)malloc(32+1);

   if ( pointer == NULL )
      exit(EXIT_FAILURE);

   for ( c = 31 ; c >= 0 ; c-- )
   {
      d = n >> c;

      if ( d & 1 )
         *(pointer+count) = 1 + '0';
      else
         *(pointer+count) = 0 + '0';

      count++;
   }
   *(pointer+count) = '\0';

   return  pointer;
}
```

Q.19 c program to reverse a number

C Program to reverse a number :- This program reverse the number entered by the user and then prints the reversed number on the screen. For example if user enter 123 as input then 321 is printed as output. In our program we use modulus(%) operator to obtain the digits of a number. To invert number look at it and write it from opposite direction or the output of code is a number obtained by writing original number from right to left. To reverse large numbers use long data type or long long data type if your compiler supports it, if you still have large numbers then use strings or other data structure.

C programming code

```
#include <stdio.h>

main()
{
   int n, reverse = 0;

   printf("Enter a number to reverse\n");
   scanf("%d",&n);

   while (n != 0)
   {
      reverse = reverse * 10;
      reverse = reverse + n%10;
      n = n/10;
   }

   printf("Reverse of entered number is = %d\n", reverse);

   return 0;
}
```

Q.20 c program to swap two numbers

C program to swap two numbers with and without using third variable, swapping in c using pointer function call by reference and using bitwise XOR operator, swapping means interchanging. For example if in your c program you have taken two variable a and b where a = 4 and b = 5, then before swapping a = 4, b = 5 after swapping a = 5, b = 4
In our c program to swap numbers we will use a temp variable to swap two numbers. Swapping is used in sorting that is when we wish to arrange numbers in a particular order either in ascending order or in descending order.

Swapping of two numbers in c

```
#include <stdio.h>

int main()
{
   int x, y, temp;

   printf("Enter the value of x and y\n");
   scanf("%d%d", &x, &y);

   printf("Before Swapping\nx = %d\ny = %d\n",x,y);

   temp = x;
   x = y;
   y = temp;

   printf("After Swapping\nx = %d\ny = %d\n",x,y);

   return 0;
}
```

Q.21 Swapping of two numbers without third variable

You can also swap two numbers *without* using temp or temporary or third variable. In that case c program will be as shown :-

```
#include <stdio.h>

int main()
{
   int a, b;

   printf("Enter two integers to swap\n");
   scanf("%d%d", &a, &b);

   a = a + b;
   b = a - b;
   a = a - b;

   printf("a = %d\nb = %d\n",a,b);
   return 0;
}
```

To understand above logic simply choose a as 7 and b as 9 and then do what is written in program. You can choose any other combination of numbers as well. Sometimes it's a good way to understand a program.

Q.22 Swap two numbers using pointers

```
#include <stdio.h>

int main()
{
   int x, y, *a, *b, temp;

   printf("Enter the value of x and y\n");
   scanf("%d%d", &x, &y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   a = &x;
   b = &y;

   temp = *b;
   *b = *a;
   *a = temp;

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}
```

Q.23 Swapping numbers using call by reference

In this method we will make a function to swap numbers.

```
#include <stdio.h>

void swap(int*, int*);

int main()
{
   int x, y;

   printf("Enter the value of x and y\n");
   scanf("%d%d",&x,&y);

   printf("Before Swapping\nx = %d\ny = %d\n", x, y);

   swap(&x, &y);

   printf("After Swapping\nx = %d\ny = %d\n", x, y);

   return 0;
}

void swap(int *a, int *b)
{
   int temp;
   temp = *b;
   *b = *a;
   *a = temp;
}
```

Q.24 C programming code to swap using bitwise XOR

```
#include <stdio.h>

int main()
{
  int x, y;

  scanf("%d%d", &x, &y);

  printf("x = %d\ny = %d\n", x, y);

  x = x ^ y;
  y = x ^ y;
  x = x ^ y;

  printf("x = %d\ny = %d\n", x, y);

  return 0;
}
```

Q.25 c program examples

Example 1 - C hello world program
/* A very simple c program printing a string on screen*/

```
#include <stdio.h>

main()
{
    printf("Hello World\n");
    return 0;
}
```

Output of above program:
"Hello World"

Example 2 - c program to take input from user using scanf

```
#include <stdio.h>

main()
{
   int number;
   printf("Enter an integer\n");
   scanf("%d",&number);

   printf("Integer entered by you is %d\n", number);

   return 0;
}
```

Output:
Enter a number
5
Number entered by you is 5

Q.26 - using if else control instructions

```
#include <stdio.h>

main()
{
   int x = 1;

   if ( x == 1 )
      printf("x is equal to one.\n");
   else
      printf("For comparison use == as = is the assignment
operator.\n");

   return 0;
}
```

Output:
x is equal to one.

Q.27 - loop example

```
#include <stdio.h>

main()
{
   int value = 1;

   while(value<=3)
   {
      printf("Value is %d\n", value);
      value++;
   }

   return 0;
}
```

Output:
Value is 1
Value is 2
Value is 3

Q.28 - c program for prime number

```
#include <stdio.h>

main()
{
   int n, c;

   printf("Enter a number\n");
   scanf("%d", &n);

   if ( n == 2 )
      printf("Prime number.\n");
   else
   {
       for ( c = 2 ; c <= n - 1 ; c++ )
       {
           if ( n % c == 0 )
              break;
       }
       if ( c != n )
          printf("Not prime.\n");
       else
          printf("Prime number.\n");
   }
   return 0;
}
```

Q.29 - command line arguments

```
#include <stdio.h>

main(int argc, char *argv[])
{
   int c;

   printf("Number of command line arguments passed: %d\n", argc);

   for ( c = 0 ; c < argc ; c++)
      printf("%d. Command line argument passed is %s\n", c+1, argv[c]);

   return 0;
}
```

Above c program prints the number and all arguments which are passed to it.

Q.30 - Array program

```
#include <stdio.h>

main()
{
    int array[100], n, c;

    printf("Enter the number of elements in array\n");
    scanf("%d", &n);

    printf("Enter %d elements\n", n);

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

    printf("Array elements entered by you are:\n");

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

    return 0;
}
```

Q.31 - function program

```
#include <stdio.h>

void my_function();

main()
{
   printf("Main function.\n");

   my_function();

   printf("Back in function main.\n");

   return 0;
}

void my_function()
{
   printf("Welcome to my function. Feel at home.\n");
}
```

Q.32 - Using comments in a program

```
#include <stdio.h>

main()
{
   // Single line comment in c source code

   printf("Writing comments is very useful.\n");

   /*
    * Multi line comment syntax
    * Comments help us to understand code later easily.
    * Will you write comments while developing programs ?
    */

   printf("Good luck c programmer.\n");

   return 0;
}
```

Q.33 - using structures in c programming

```
#include <stdio.h>

struct programming
{
    float constant;
    char *pointer;
};

main()
{
   struct programming variable;
   char string[] = "Programming in Software Development.";

   variable.constant = 1.23;
   variable.pointer = string;

   printf("%f\n", variable.constant);
   printf("%s\n", variable.pointer);

   return 0;
}
```

Q.34 - c program for Fibonacci series

```
#include <stdio.h>

main()
{
   int n, first = 0, second = 1, next, c;

   printf("Enter the number of terms\n");
   scanf("%d",&n);

   printf("First %d terms of Fibonacci series are :-\n",n);

   for ( c = 0 ; c < n ; c++ )
   {
      if ( c <= 1 )
         next = c;
      else
      {
         next = first + second;
         first = second;
         second = next;
      }
      printf("%d\n",next);
   }

   return 0;
}
```

Q.35- c graphics programming

```
#include <graphics.h>
```

```
#include <conio.h>

main()
{
    int gd = DETECT, gm;

    initgraph(&gd, &gm,"C:\\TC\\BGI");

    outtextxy(10,20, "Graphics source code example.");

    circle(200, 200, 50);

    setcolor(BLUE);

    line(350, 250, 450, 50);

    getch();
    closegraph( );
    return 0;
}
```

For GCC user

If you are using GCC on Linux operating system then you need to modify programs. For example consider the following program which prints first ten natural numbers

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int c;

    for ( c = 1 ; c <= 10 ; c++ )
        printf("%d\n", c);

    getch();
    return 0;
}
```

Above source code includes a header file <conio.h> and uses function getch, but this file is Borland specific so it works in turbo c compiler but not in GCC. So the code for GCC should be like

```
#include <stdio.h>

int main()
{
    int c;

    /* for loop */

    for ( c = 1 ; c <= 10 ; c++ )
        printf("%d\n", c);

    return 0;
}
```

Q.36 c program to print patterns of numbers and stars

These program prints various different patterns of numbers and stars. These codes illustrate how to create various patterns using c programming. Most of these c programs involve usage of nested loops and space. A pattern of numbers, star or characters is a way of arranging these in some logical manner or they may form a sequence. Some of these patterns are triangles which have special importance in mathematics. Some patterns are symmetrical while other are not. Please see the complete page and look at comments for many different patterns.

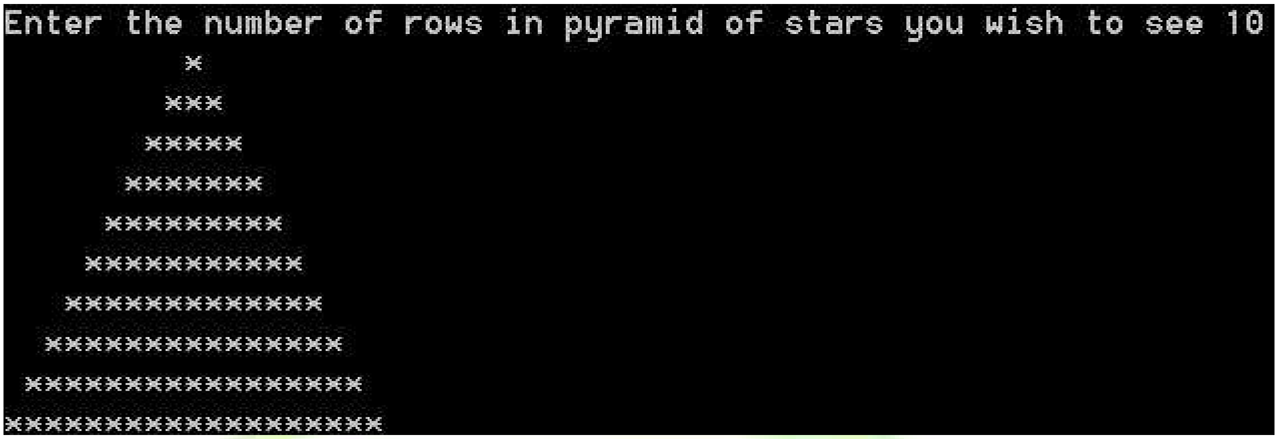
```
    *
   ***
  *****
 *******
*********
```

We have shown five rows above, in the program you will be asked to enter the numbers of rows you want to print in the pyramid of stars.

C programming code

```
#include<stdio.h>

main()
{
   int row, c, n, temp;

   printf("Enter the number of rows in pyramid of stars you wish to see
");
   scanf("%d",&n);

   temp = n;

   for ( row = 1 ; row <= n ; row++ )
   {
      for ( c = 1 ; c < temp ; c++ )
         printf(" ");

      temp--;

      for ( c = 1 ; c <= 2*row - 1 ; c++ )
         printf("*");

      printf("\n");
   }

   return 0;
}
```

Output:



```
Enter the number of rows in pyramid of stars you wish to see 10
         *
        ***
       *****
      *******
     *********
    ***********
   *************
  ***************
 *****************
*******************
```

Consider the pattern

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

to print above pattern see the code below:

```
#include<stdio.h>

main()
{
    int n, c, k;

    printf("Enter number of rows\n");
    scanf("%d",&n);

    for ( c = 1 ; c <= n ; c++ )
    {
        for( k = 1 ; k <= c ; k++ )
            printf("*");

        printf("\n");
    }

    return 0;
}
```

Q.37c program to find maximum element in array

This code find maximum or largest element present in an array. It also prints the location or index at which maximum element occurs in array. This can also be done by using pointers (see both codes).

C programming code

```
#include <stdio.h>

int main()
{
  int array[100], maximum, size, c, location = 1;

  printf("Enter the number of elements in array\n");
  scanf("%d", &size);

  printf("Enter %d integers\n", size);

  for (c = 0; c < size; c++)
    scanf("%d", &array[c]);

  maximum = array[0];

  for (c = 1; c < size; c++)
  {
    if (array[c] > maximum)
    {
       maximum  = array[c];
       location = c+1;
    }
  }

  printf("Maximum element is present at location number %d and it's
value is %d.\n", location, maximum);
  return 0;
}
```

Q.38 C programming code using pointers

```
#include <stdio.h>

int main()
{
  long array[100], *maximum, size, c, location = 1;

  printf("Enter the number of elements in array\n");
  scanf("%ld", &size);

  printf("Enter %ld integers\n", size);

  for ( c = 0 ; c < size ; c++ )
    scanf("%ld", &array[c]);

  maximum  = array;
  *maximum = *array;

  for (c = 1; c < size; c++)
  {
    if (*(array+c) > *maximum)
    {
       *maximum = *(array+c);
       location = c+1;
    }
  }

  printf("Maximum element is present at location number %ld and it's
value is %ld.\n", location, *maximum);
  return 0;
}
```

Q.38c program to insert an element in an array

This code will insert an element into an array, For example consider an array a[10] having three elements in it initially and a[0] = 1, a[1] = 2 and a[2] = 3 and you want to insert a number 45 at location 1 i.e. a[0] = 45, so we have to move elements one step below so after insertion a[1] = 1 which was a[0] initially, and a[2] = 2 and a[3] = 3. Array insertion does not mean increasing its size i.e array will not be containing 11 elements.

C programming code

```
#include <stdio.h>

int main()
{
   int array[100], position, c, n, value;

   printf("Enter number of elements in array\n");
   scanf("%d", &n);

   printf("Enter %d elements\n", n);

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

   printf("Enter the location where you wish to insert an element\n");
   scanf("%d", &position);

   printf("Enter the value to insert\n");
   scanf("%d", &value);

   for (c = n - 1; c >= position - 1; c--)
      array[c+1] = array[c];

   array[position-1] = value;

   printf("Resultant array is\n");

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

   return 0;
}
```

Q.39C program to shutdown or turn off computer

C Program to shutdown your computer: This program turn off i.e shutdown your computer system. Firstly it will asks you to shutdown your computer if you press 'y' the your computer will shutdown in 30 seconds, system function of "stdlib.h" is used to run an executable file shutdown.exe which is present in C:\WINDOWS\system32 in Windows XP. You can use various options while executing shutdown.exe for example -s option shutdown the computer after 30 seconds, if you wish to shutdown immediately then you can write "shutdown -s -t 0" as an argument to system function. If you wish to restart your computer then you can write "shutdown -r".

If you are using Turbo C Compiler then execute your file from folder. Press F9 to build your executable file from source program. When you run from within the compiler by pressing Ctrl+F9 it may not work.

C programming code for Windows XP

```
#include <stdio.h>
#include <stdlib.h>

main()
{
   char ch;

   printf("Do you want to shutdown your computer now (y/n)\n");
   scanf("%c",&ch);

   if (ch == 'y' || ch == 'Y')
      system("C:\\WINDOWS\\System32\\shutdown -s");

   return 0;
}
```

C programming code for Windows 7

```
#include <stdio.h>
#include <stdlib.h>

main()
{
   char ch;

   printf("Do you want to shutdown your computer now (y/n)\n");
   scanf("%c",&ch);

   if (ch == 'y' || ch == 'Y')
      system("C:\\WINDOWS\\System32\\shutdown /s");

   return 0;
}
```

To shutdown immediately use "C:\\WINDOWS\\System32\\ shutdown /s /t 0". To restart use /r instead of /s.

C programming code for Ubuntu Linux

```
#include <stdio.h>

int main() {
  system("shutdown -P now");
  return 0;
}
```

You need to be logged in as **root** user for above program to execute otherwise you will get the message *shutdown: Need to be root*, **now** specifies that you want to shutdown immediately. '-P' option specifes you want to power off your machine. You can specify minutes as:
shutdown -P "number of minutes"
For more help or options type at terminal: man shutdown.

Q 40c program to read a file

C program to read a file :- This program reads a file entered by the user and displays it's contents on the screen, fopen function is used to open a file, it returns a pointer to structure FILE. FILE is a predefined structure in stdio.h . If the file is successfully opened then fopen returns a pointer to file and if it is unable to open a file then it returns NULL. fgetc function returns a character which is read from the file and fclose function closes the file. Opening a file means we bring file from disk to ram to perform operations on it. The file must be present in the directory in which the executable file of this code sis present.

C program to open a file

C programming code to open a file and to print it contents on screen.

```
#include<stdio.h>
#include<stdlib.h>

main()
{
   char ch, file_name[25];
   FILE *fp;

   printf("Enter the name of file you wish to see ");
   gets(file_name);

   fp = fopen(file_name,"r"); // read mode

   if( fp == NULL )
   {
      perror("Error while opening the file.\n");
      exit(EXIT_FAILURE);
   }

   printf("The contents of %s file are :- \n\n", file_name);

   while( ( ch = fgetc(fp) ) != EOF )
      printf("%c",ch);

   fclose(fp);
   return 0;
}
```