

8/27/2011

C

C PROGRAMMING QUESTIONS AND ANSWER

Find out the perfect number using c program

```
#include<stdio.h>
int main() {
    int n,i=1,sum=0;
    printf("\nEnter a number:-");
    scanf("%d",&n);
    while(i<n) {
        if(n%i==0)
            sum=sum+i;
        i++;
    }
    if(sum==n)
        printf("\nThe no %d is a perfect number",i);
    else
        printf("\nThe no %d is not a perfect number",i);
    return 0;
}
```

Check the given number is Armstrong number or not using c program

```
#include<stdio.h>
int main() {
    int num,r,sum=0,temp;
    printf("\nEnter a number:-");
    scanf("%d",&num);
    temp=num;
    while(num!=0) {
        r=num%10;
        num=num/10;
        sum=sum+(r*r*r);
    }
    if(sum==temp)
        printf("%d is an Armstrong number",temp);
    else
        printf("%d is not an Armstrong number",temp);
}
```

```

sum=sum+(r*r*r);

    }

    if(sum==temp)

        printf("\nThe number %d is an armstrong
number",temp);

    else

        printf("\nThe number %d is not an armstrong
number",temp);

    return 0;

}

```

Definition of Armstrong number:

Definition for c programming point of view:

Those numbers which sum of the cube of its digits is equal to that number are known as Armstrong numbers. For example 153 since $1^3 + 5^3 + 3^3 = 1 + 125 + 9 = 153$

Other Armstrong numbers: 370,371,407 etc.

In general definition:

Those numbers which sum of its digits to power of number of its digits is equal to that number are known as Armstrong numbers.

Example 1: 153

Total digits in 153 is 3

And $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

Example 2: 1634

Copyright@ritesh kumar: <http://cquestionbank.blogspot.com/>

Total digits in 1634 is 4

And $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 64 = 1634$

Examples of Armstrong numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725

Check given number is prime number or not using c program

```
#include<stdio.h>

int main(){

    int num,i,count=0;
    printf("\nEnter a number:");
    scanf("%d",&num);
    for(i=2;i<=num/2;i++){
        if(num%i==0){
            count++;
            break;
        }
    }
    if(count==0)
        printf("%d is a prime number",num);
    else
        printf("%d is not a prime number",num);
    return 0;
}
```

Definition of prime number:

A natural number greater than one has not any other divisors except 1 and itself. In other word we can say which has only two divisors 1 and number itself. For example: 5

Their divisors are 1 and 5.

Note: 2 is only even prime number.

Example of prime numbers : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199 etc.

Reverse any number using c program

```
#include<stdio.h>
int main() {
    int num,out;
    scanf("%d",&num);
    out=rev(num);
    printf("%d",out);
    return 0;
}

int rev(int num) {
    static sum,r;
    if(num) {
        r=num%10;
        sum=sum*10+r;
        rev(num/10);
    }
    else return 0;
    return sum;
}
```

Write a c program to check given number is strong number or not

```
#include<stdio.h>
int main() {
    int num,i,f,r,sum=0,temp;
    printf("\nEnter a number");
    scanf("%d",&num);
    temp=num;
```

```

while(num) {
    i=1, f=1;
    r=num%10;
    while(i<=r) {
        f=f*i;
        i++;
    }
    sum=sum+f;
    num=num/10;
}
if(sum==temp)
    printf("%d is a strong number",temp);
else
    printf("%d is not a strong number",temp);
return 0;
}

```

Definition of strong number:

A number is called strong number if sum of the factorial of its digit is equal to number itself. For example: 145 since
 $1! + 4! + 5! = 1 + 24 + 120 = 145$

Write a c program to find out sum of digit of given number

```

#include<stdio.h>
int main() {
    int num, sum=0, r;
    printf("\nEnter a number:");
    scanf("%d", &num);
    while(num) {
        r=num%10;
        num=num/10;
        sum=sum+r;
    }
    printf("sum=%d", sum);
    return 0;
}

```

Copyright@ritesh kumar: <http://cquestionbank.blogspot.com/>

```
}
```

Check the given number is palindrome number or not using c program

```
#include<stdio.h>
int main() {
    int num,r,sum=0,temp;
    printf("\nEnter a number:");
    scanf("%d",&num);
    temp=num;
    while(num) {
        r=num%10;
        num=num/10;
        sum=sum*10+r;
    }
    if(temp==sum)
        printf("\n%d is a palindrome",temp);
    else
        printf("\n%d is not a palindrome",temp);
    return 0;
}
```

Definition of Palindrome number:

A number is called palindrome number if it is remain same when its digits are reversed. For example 121 is palindrome number. When we will reverse its digit it will remain same number i.e. 121

Examples of palindrome number: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, 131, 141, 151, 161, 171, 181, 191 etc.

Find out generic root of a number by c program

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

Copyright@ritesh kumar: <http://cquestionbank.blogspot.com/>

```

    long int num, sum, r;
    printf("\nEnter a number:-");
    scanf("%ld", &num);
    while (num > 10) {
        sum = 0;
        while (num) {
            r = num % 10;
            num = num / 10;
            sum += r;
        }
        if (sum > 10)
            num = sum;
        else
            break;
    }
    printf("\nSum of the digits in single digit is:
    %ld", sum);
    return 0;
}

```

Find factorial of a number using c program

```

#include<stdio.h>
int main() {
    int i=1, f=1, num;
    printf("\nEnter a number:");
    scanf("%d", &num);
    while (i <= num) {
        f = f * i;
        i++;
    }
    printf("\nFactorial of %d is:%d", num, f);
    return 0;
}

```

Checking leap year using c program

```

#include<stdio.h>

```

Copyright@ritesh kumar: <http://cquestionbank.blogspot.com/>


```

#include<conio.h>
void main(){
    int year;
    clrscr();
    printf("Enter any year: ");
    scanf("%d",&year);
    if(((year%4==0)&&(year%100!=0))||(year%400==0))
        printf("%d is a leap year",year);
    else
        printf("%d is not a leap year",year);
    getch();
}

```

Definition of leap year:

Rule 1: A year is called leap year if it is divisible by 400.

For example: 1600, 2000 etc leap year while 1500, 1700 are not leap year.

Rule 2: If year is not divisible by 400 as well as 100 but it is divisible by 4 then that year are also leap year.

For example: 2004, 2008, 1012 are leap year.

Algorithm of leap year:

```

IF year MODULER 400 IS 0
    THEN leap_year
ELSE IF year MODULER 100 IS 0
    THEN not_leap_year
ELSE IF year MODULER 4 IS 0
    THEN leap_year
ELSE
    not_leap_year

```

Write a c program to find out L.C.M. of two numbers.

```

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

```

```

int n1,n2,x,y;
printf("\nEnter two numbers:");
scanf("%d %d",&n1,&n2);
x=n1,y=n2;
while(n1!=n2){
    if(n1>n2)
        n1=n1-n2;
    else
        n2=n2-n1;
}
printf("L.C.M=%d",x*y/n1);
return 0;
}

```

Definition of LCM (Least common multiple):

LCM of two integers is a smallest positive integer which is multiple of both integers that it is divisible by the both of the numbers.

For example: LCM of two integers 2 and 5 is 10 since 10 is the smallest positive numbers which is divisible by both 2 and 5.

Swap two variables without using third using c program variable

```

#include<stdio.h>
int main(){
    int a,b;
    printf("\nEnter two numbers:");
    scanf("%d %d",&a,&b);
    printf("\nBefore swapping a=%d b=%d",a,b);
    a=a^b;
    b=b^a;
    a=a^b;
    printf("\nAfter swapping a=%d b=%d",a,b);
    return 0;
}

```

OR

Swapping of two number

```
#include<stdio.h>
int main() {
    int a=5,b=10;
    //process one
    a=b+a;
    b=a-b;
    a=a-b;
    printf("a= %d  b=  %d",a,b);

    //process two
    a=5;
    b=10;
    a=a+b-(b=a);
    printf("\na= %d  b=  %d",a,b);
    //process three
    a=5;
    b=10;
    a=a^b;
    b=a^b;
    a=b^a;
    printf("\na= %d  b=  %d",a,b);

    //process four
    a=5;
    b=10;
    a=b-~a-1;
    b=a+~b+1;
    a=a+~b+1;
    printf("\na= %d  b=  %d",a,b);

    //process five
    a=5,
    b=10;
    a=b+a,b=a-b,a=a-b;
    printf("\na= %d  b=  %d",a,b);
    getch();
}
```

Copyright@ritesh kumar: <http://cquestionbank.blogspot.com/>

```
}
```

Program to convert decimal to binary in c

```
#include<stdio.h>

int main() {
    long int decimalNumber,remainder,quotient;
    int binaryNumber[100],i=1,j;

    printf("Enter any decimal number: ");
    scanf("%ld",&decimalNumber);

    quotient = decimalNumber;

    while(quotient!=0){
        binaryNumber[i++]= quotient % 2;
        quotient = quotient / 2;
    }

    printf("Equivalent binary value of decimal number
%d: ",decimalNumber);
    for(j = i -1 ;j> 0;j--)
        printf("%d",binaryNumber[j]);

    return 0;
}
```

Sample output:

```
Enter any decimal number: 50
Equivalent binary value of decimal number 50: 110010
```

Algorithm:

Binary number system: It is base 2 number system which uses the digits from 0 and 1.

Decimal number system:

It is base 10 number system which uses the digits from 0 to 9

Convert from decimal to binary algorithm:

Following steps describe how to convert decimal to binary

Step 1: Divide the original decimal number by 2

Step 2: Divide the quotient by 2

Step 3: Repeat the step 2 until we get quotient equal to zero.

Equivalent binary number would be remainders of each step in the reverse order.

Decimal to binary conversion with example:

For example we want to convert decimal number 25 in the binary.

```
Step 1:  25 / 2  Remainder : 1 , Quotient : 12
Step 2:  12 / 2  Remainder : 0 , Quotient : 6
Step 3:   6 / 2  Remainder : 0 , Quotient : 3
Step 4:   3 / 2  Remainder : 1 , Quotient : 1
Step 5:   1 / 2  Remainder : 1 , Quotient : 0
```

So equivalent binary number is: 11001

That is $(25)_{10} = (11001)_2$

Multiplication of two matrices using c program

```
#include<stdio.h>
int main(){
    int a[5][5],b[5][5],c[5][5],i,j,k,sum=0,m,n,o,p;
    printf("\nEnter the row and column of first matrix");
    scanf("%d %d",&m,&n);
```

```

printf("\nEnter the row and column of second
matrix");
scanf("%d %d",&o,&p);
if(n!=o){
    printf("Matrix multiplication is not possible");
    printf("\nColumn of first matrix must be same as
row of second matrix");
}
else{
    printf("\nEnter the First matrix->");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++){
            scanf("%d",&a[i][j]);
        }
    printf("\nEnter the Second matrix->");
    for(i=0;i<o;i++)
        for(j=0;j<p;j++){
            scanf("%d",&b[i][j]);
        }
    printf("\nThe First matrix is\n");
    for(i=0;i<m;i++){
        printf("\n");
        for(j=0;j<n;j++){
            printf("%d\t",a[i][j]);
        }
    }
    printf("\nThe Second matrix is\n");
    for(i=0;i<o;i++){
        printf("\n");
        for(j=0;j<p;j++){
            printf("%d\t",b[i][j]);
        }
    }
    for(i=0;i<m;i++)
        for(j=0;j<p;j++){
            c[i][j]=0;
        }
    for(i=0;i<m;i++){ //row of first matrix
        for(j=0;j<p;j++){ //column of second matrix
            sum=0;
            for(k=0;k<n;k++){
                sum=sum+a[i][k]*b[k][j];
            }
            c[i][j]=sum;
        }
    }
}

```

```

    }
    }
}
printf("\nThe multiplication of two matrix is\n");
for(i=0;i<m;i++){
    printf("\n");
    for(j=0;j<p;j++){
        printf("%d\t",c[i][j]);
    }
}
return 0;
}

```

Algorithm:

Multiplication of two matrixes:

Rule: Multiplication of two matrixes is only possible if first matrix has size $m \times n$ and other matrix has size $n \times r$. Where m , n and r are any positive integer.

Multiplication of two matrixes is defined as

$$[AB]_{i,j} = \sum_{s=1}^n A_{i,s} B_{s,j}$$

Where $1 \leq i \leq m$ and $1 \leq j \leq n$

For example:

Suppose two matrixes A and B of size of 2×2 and 2×3 respectively:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

Create a file and store data in it in c program

```
#include<stdio.h>
int main() {
    FILE *fp;
    char ch;
    fp=fopen("file.txt","w");
    printf("\nEnter data to be stored in to the
file:");
    while ((ch=getchar()) != EOF)
        putc(ch,fp);
    fclose(fp);
    return 0;
}
```


Find factorial of a number using recursion in c program

```
#include<stdio.h>
int main() {
    int num, f;
    printf("\nEnter a number: ");
    scanf("%d", &num);
    f=fact(num);
    printf("\nFactorial of %d is: %d", num, f);
    return 0;
}

int fact(int n) {
    if(n==1)
        return 1;
    else
        return (n*fact(n-1));
}
```