What is the output?

#include <iostream>

using namespace std;

int main(){

int j=0;

for (int i=0;i<100;i++)

{

j= j++;

}

cout<< j<< endl;

return 0;

}

0

What will be output of following c program?

**#include**<stdio.h>

**int** main(){

**enum** number { a=-1, b= 4,c,d,e};

printf("%d",e);

**return** 0;

}

7

#include<stdio.h>

int main(){

int i=0;

for(i=0;i<20;i++){

switch(i){

case 0:i+=5;

case 1:i+=2;

case 5:i+=5;

default: i+=4; break;

}

printf("%d ",i);

}

return 0;

}

16 21

**#include**<stdio.h>

**int** main(){

**char** c=-64;

**int** i=-32;

**unsigned** **int** u =-16;

**if**(c>i){

printf("pass1");

**if**(c<i)

printf("pass2");

**else**

printf("Fail2");

}

**else**

printf("Fail1”);

**if**(c==i)

printf("pass2");

**else**

printf("Fail2");

**return** 0;

}

Fail1Fail2

**#include**<stdio.h>

**#include**<string.h>

**char** \*gxxx(){

**static** **char** xxx[1024];

**return** xxx;

}

**int main**(){

**char** \*g="string";

strcpy(gxxx(),g);

g = gxxx();

strcpy(g,"oldstring");

printf("The string is : %s",gxxx());

**return** 0;

}

The string is: Oldstring

Which of the choices is true for the mentioned declaration?

**const** **char** \*p;

and

**char** \* **const** p;

In first case, you can't change the character and second case you can’t change the pointer

#include<stdio.h>

#include<string.h>

#include<conio.h>

int main()

{

struct field

{

int a;

char b;

};

struct field bit1={5,'A'};

char \*p;

p= (char \*) &bit1;

\*p=45; // replace 45 with 260

printf("\n%d",bit1.a);

getch();

}

45

#include<stdio.h>

#include<string.h>

#include<conio.h>

int main(){

int a = 320;

char \*ptr;

ptr =( char \*)&a;

printf("%d ",\*ptr);

return 0;

}

As we know int is two byte data byte while char is one byte data byte. char pointer can keep the address one byte at time.

Binary value of 320 is 00000001 01000000 (In 16 bit)

So ptr is pointing only first 8 bit which color is green and Decimal value is 64.

64

#include<stdio.h>

int main(){

   int i = 3;

   int \*j;

   int \*\*k;

   j=&i;

   k=&j;

   printf("%u %u %d ",k,\*k,\*\*k);  
   return 0;

}

|  |
| --- |
| Address, Address, 3 |

#include<stdio.h>

#include<string.h>

int main(){

int a = 5,b = 10,c;

int \*p = &a,\*q = &b;

c = p - q;

printf("%d %d " , p, q);

printf("%d" , c);

return 0;

}

Difference of two same type of pointer is always one

Pattern generation

#include<stdio.h>

void pattern(int);

int main()

{

int n;

printf("Enter n (rows/columns) value");

scanf("%d",&n);

pattern(n);

return 0;

}

void pattern(int n)

{

int i,j,k;

for(i=1;i<=n;i++)

{

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

{

printf("%d",i);

}

printf("\n");

}

for(i=n;i>=1;i--)

{

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

{

printf("%d",i);

}

printf("\n");

}

}

1

2 2

3 3 3

4 4 4 4

4 4 4 4

3 3 3

2 2

1

Just replace i with j in the printf statement and you will get the pattern as

1

1 2

1 2 3

1 2 3 4

1 2 3 4

1 2 3

1 2

1

Floyd triangle

#include<stdio.h>

void pattern(int);

int main()

{

int n;

printf("Enter n (rows/columns) value");

scanf("%d",&n);

pattern(n);

return 0;

}

void pattern(int n)

{

int i,j,k=1;

for(i=1;i<=n;i++)

{

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

{

printf("%d ",k);

k++;

}

printf("\n");

}

}

1

2 3

4 5 6

7 8 9 10

Triangle with stars

#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 spacing purpose \*/

for ( c = 1 ; c < temp ; c++ )

printf(" ");

temp--;

/\* for printing stars \*/

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

printf("\*");

printf("\n");

}

}



* Printing stars with left aligned by removing the for loop which places the stars.
* Printing stars with right aligned by replacing temp in the for loop with 2\*temp-1 which places the stars.
* To print inverted pyramid,
  + for ( row = 1 ; row <= n ; row++ ) and
  + change temp = 1 and temp--.

Alphabets

#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);

char alphabet = 'A';

for ( row = 1 ; row <= n ; row++ )

{

/\* for printing stars \*/

for ( c = 1 ; c <= row ; c++ )

printf("%c", alphabet);

++alphabet;

printf("\n");

}

}

A

B B

C C C

D D D D

E E E E E

#include<stdio.h>

main()

{

int row, c, n, temp;

printf("Enter the number of rows in pyramid of stars you wish to see "); // give input in ASCI

scanf("%d",&n);

char alphabet = 'A';

temp=n-64;

for ( row = 1 ; row <= n-'A'+1 ; row++ ) //if input is given in ASCII

{

/\* for spacing purpose \*/

for ( c = 1 ; c < temp ; c++ )

printf(" ");

temp--;

/\* for printing \*/

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

printf("%c", alphabet);

++alphabet;

printf("\n");

}

}

A

B B

C C C

D D D D

E E E E E

Factorial with recursion

#include <stdio.h>

long int multiplyNumbers(int n);

int main()

{

int n;

printf("Enter a positive integer: ");

scanf("%d", &n);

printf("Factorial of %d = %ld", n, multiplyNumbers(n));

return 0;

}

long int multiplyNumbers(int n)

{

if (n >= 1)

return n\*multiplyNumbers(n-1);

else

return 1;

}

**Whether Armstrong number or not ? (Length = 3)**

#include <stdio.h>

int main()

{

int number, originalNumber, remainder, result = 0;

printf("Enter a three digit integer: ");

scanf("%d", &number);

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber%10;

result += remainder\*remainder\*remainder;

originalNumber /= 10;

}

if(result == number)

printf("%d is an Armstrong number.",number);

else

printf("%d is not an Armstrong number.",number);

return 0;

}

**Whether Armstrong number or not ? (Length=n)**

#include <stdio.h>

#include <math.h>

int main()

{

int number, originalNumber, remainder, result = 0, n = 0 ;

printf("Enter an integer: ");

scanf("%d", &number);

originalNumber = number;

while (originalNumber != 0)

{

originalNumber /= 10;

++n;

}

originalNumber = number;

while (originalNumber != 0)

{

remainder = originalNumber%10;

result += pow(remainder, n);

originalNumber /= 10;

}

if(result == number)

printf("%d is an Armstrong number.", number);

else

printf("%d is not an Armstrong number.", number);

return 0;

}

Enter a three digit integer: 371

371 is an Armstrong number. Eg: 371=(3^3)+(7^3)+(1^3)=371

Enter an integer: 1634

1634 is an Armstrong number.

**Bubble sort**

#include <stdio.h>

int main()

{

int data[100],i,n,step,temp;

printf("Enter the number of elements to be sorted: ");

scanf("%d",&n);

for(i=0;i<n;++i)

{

printf("%d. Enter element: ",i+1);

scanf("%d",&data[i]);

}

for(step=0;step<n-1;++step)

for(i=0;i<n-step-1;++i)

{

if(data[i]>data[i+1]) /\* To sort in descending order, change > to < in this line. \*/

{

temp=data[i];

data[i]=data[i+1];

data[i+1]=temp;

}

}

printf("In ascending order: ");

for(i=0;i<n;++i)

printf("%d ",data[i]);

return 0;

}

**Fibonacci series :**

#include<stdio.h>

int main(){

int i,range;

long int arr[40];

printf("Enter the number range: ");

scanf("%d",&range);

arr[0]=0;

arr[1]=1;

for(i=2;i<range;i++){

arr[i] = arr[i-1] + arr[i-2];

}

printf("Fibonacci series is: ");

for(i=0;i<range;i++)

printf("%ld ",arr[i]);

return 0;

}

**Pascal Triangle**

#include<stdio.h>

long fact(int);

int main(){

int line,i,j;

printf("Enter the no. of lines: ");

scanf("%d",&line);

for(i=0;i<line;i++){

for(j=0;j<line-i-1;j++)

printf(" ");

/\* alternate for printing space

temp=line;

for ( c = 1 ; c < temp ; c++ )

printf(" ");

temp--;\*/

for(j=0;j<=i;j++)

printf("%ld ",fact(i)/(fact(j)\*fact(i-j)));

printf("\n");

}

return 0;

}

long fact(int num){

long f=1;

int i=1;

while(i<=num){

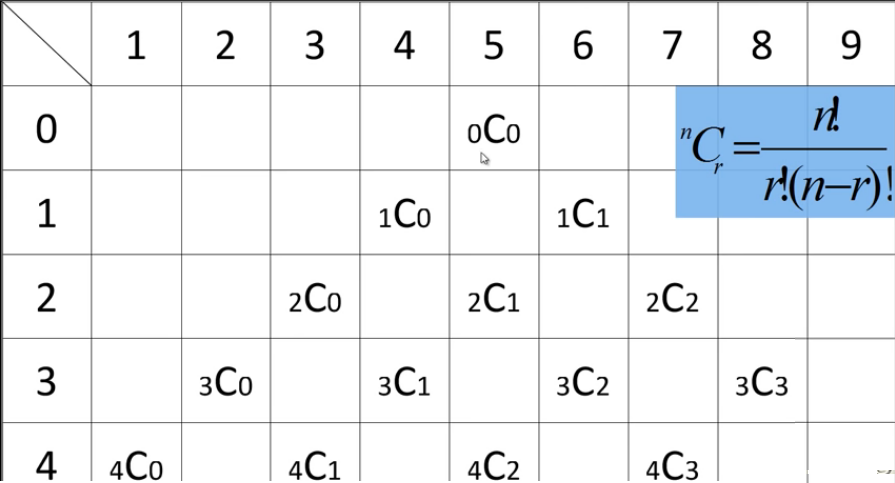
f=f\*i;

i++;

}

return f;

}



Enter the no. of lines: 8

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

**Prime number**

#include<stdio.h>

int main(){

int num,i,count=0;

printf("Enter a number: ");

scanf("%d",&num);

for(i=2;i<=num/2;i++){

if(num%i==0){

count++;

break;

}

}

if(count==0 && num!= 1)

printf("%d is a prime number",num);

else

printf("%d is not a prime number",num);

return 0;

}

**Strong Number (145 40585)**

#include<stdio.h>

int main() {

int num,i,f,r,sum=0,temp;

printf("Enter 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)

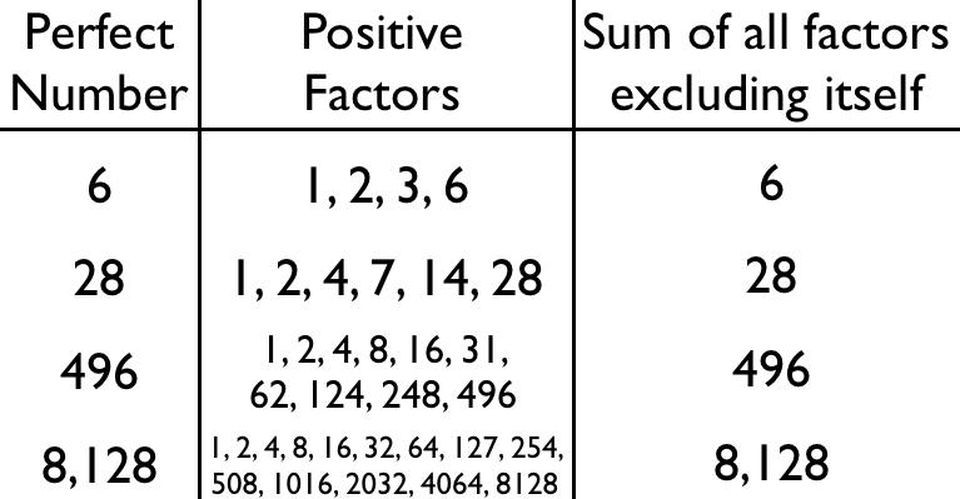
printf("%d is a strong number",temp); else

printf("%d is not a strong number",temp);

return 0;

}

**Perfect Number**



#include<stdio.h>

int main(){

int n,i=1,sum=0;

printf("Enter a number: ");

scanf("%d",&n);

while(i<n){

if(n%i==0)

sum=sum+i;

i++;

}

if(sum==n)

printf("%d is a perfect number",i);

else

printf("%d is not a perfect number",i);

return 0;

}

**Palindrome Number (141 and 11511, etc)**

#include<stdio.h>

int main(){

int num,r,sum=0,temp;

printf("Enter a number: ");

scanf("%d",&num);

temp=num;

while(num){

r=num%10;

num=num/10;

sum=sum\*10+r;

}

if(temp==sum)

printf("%d is a palindrome",temp);

else

printf("%d is not a palindrome",temp);

return 0;

}