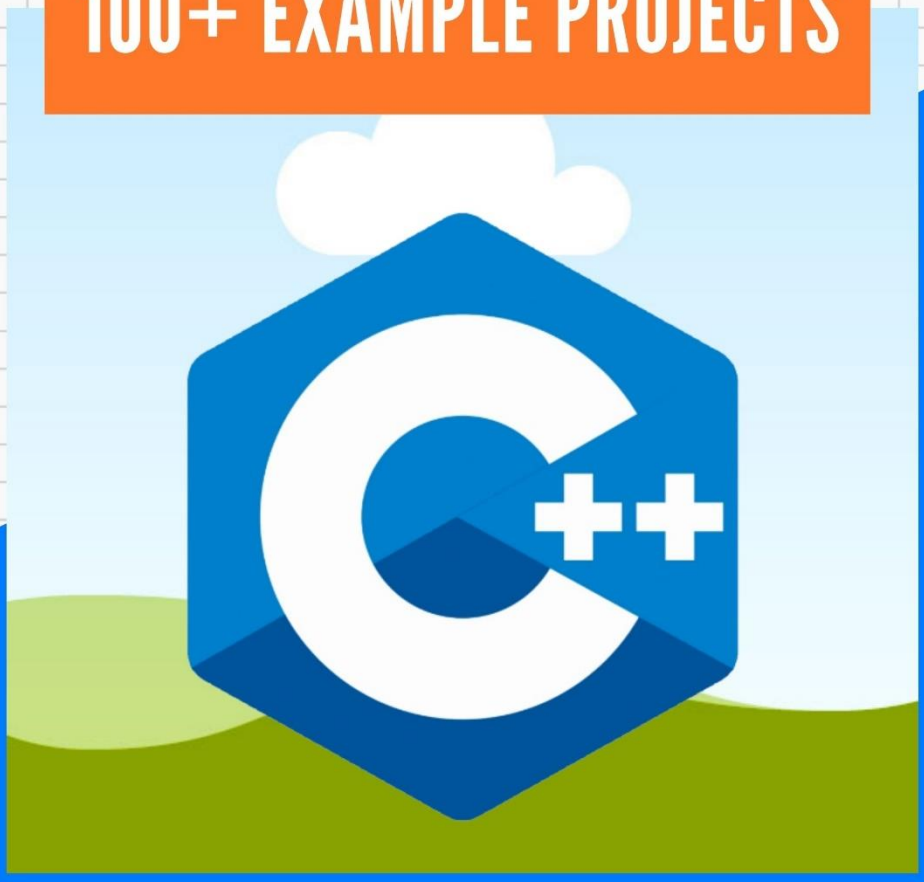
PRACTICAL C++ PROGRAMMING

100+ EXAMPLE PROJECTS



EMENWA GLOBAL

Gstevewall Academy

www.gstevewall.com

Practice 1: A Program to Print Hello World

```
#include <iostream>

using namespace std;

int main()
{
cout << "Hello World" << endl;

return 0;
}
```

Output

```
Hello World

Process returned 0 (0x0)   execution time : 0.250 s
Press any key to continue.
```

Practice 2: A Program for Use of Variables

```
#include <iostream>
#include <string>

using namespace std;

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

string hello = "Peter Paul";

x = 20;
y = 9;
z = x + y;
cout << "Use of Variables in C++" << endl;

cout << "The result of x + y = " << z << endl;

cout << "My name is " << hello << endl;
return 0;
}
```

Output

```
Use of Variables in C++
The result of x + y = 29
My name is Peter Paul

Process returned 0 (0x0)   execution time : 0.250 s
Press any key to continue.
```

Practice 3: A Program for Local and Global Variables

```
#include <iostream>

using namespace std;

int x =9, w = 6;
float z;

int main()
{
int x = 3;
float z;

cout << "Local and Global Variables in C++ \n" << endl;

cout << "The value of z = " << x + w << endl;
return 0;
}
```

Output

```
Local and Global Variables in C++

The value of z = 9

Process returned 0 (0x0)   execution time : 0.141 s
Press any key to continue.
```

Practice 4: A Program for User Input

```
#include <iostream>

using namespace std;

int main()
{
int x, y;
cout << "User Input \n" << endl;

cout << "Insert a number: " << endl;

cin >> x;

cout << "Insert second number: " << endl;

cin >> y;

cout << "You entered: " << x << " and " << y << endl;

return 0;
}
```

Output

```
User Input

Insert a number:
12
Insert second number:
4
You entered: 12 and 4

Process returned 0 (0x0)   execution time : 8.264 s
Press any key to continue.
```

Practice 5: A Program to Accept Strings from Users

```
#include <iostream>

using namespace std;

int main()
{
char STR1[200];
cout << "String Input \n" << endl;


cout << "Insert some strings: " << endl;

cin.getline(STR1,200);

cout << "You entered: " << STR1 << endl;

return 0;
}
```

Output

```
String Input

Insert some strings:
Ejike IfeanyiChukwu
You entered: Ejike IfeanyiChukwu

Process returned 0 (0x0)   execution time : 6.294 s
Press any key to continue.
```

Practice 6: A Program to Add 2 Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, y, sum;
cout << "Addition of two numbers \n" << endl;

cout << "Insert first number: " << endl;

cin >> x;

cout << "Insert second number: " << endl;

cin >> y;

sum = x + y;

cout << "The result of x + y = "<< sum << endl;

return 0;
}
```

Output

```
Addition of two numbers

Insert first number:
8
Insert second number:
5
The result of x + y = 13

Process returned 0 (0x0)   execution time : 10.585 s
Press any key to continue.
```

Practice 7: A Program to Use Math Operators

```
#include <iostream>

using namespace std;

int main()
{
int x, y, sum, mul, sub, mod;
float div;
cout << "Addition, Subtraction, Multiplication and Division of two numbers"
<< endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

sum = x + y;
mul = x * y;
sub = x - y;
mod = x % y;
div = x / y;

cout << "The sum of "<< x << " + " << y << " = " << sum <<endl;
cout << "The Multiplication of x * y = " << mul << endl;
cout << "The subtraction of x - y = " << sub << endl;
cout << "The modulus of x%y = " << mod << endl;
cout << "The division of x/y = " << div << endl;

return 0;
}
```

Output

```
Addition, Subtraction, Multiplication and Division of two numbers
Insert first number:
6
Insert second number:
4
The sum of 6 + 4 = 10
The Multiplication of x * y = 24
The subtraction of x - y = 2
The modulus of x%y = 2
The division of x/y = 1

Process returned 0 (0x0)   execution time : 10.758 s
Press any key to continue.
```

Practice 8: A Program to Add n Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, sum=0;
cout << "Adding N Numbers \n" << endl;

cout << "Please choose range of number to add: " << endl;
cin >> x;

cout << "Please insert " << x << " number(s) only: " << endl;

for(y=1; y<=x; y++)
{
cin >> z;
sum = sum + z;
}

cout << "The result of addition of " << x << " number(s) is = " << sum <<
endl;

return 0;
}
```

Output

```
Adding N Numbers

Please choose range of number to add:
4
Please insert 4 number(s) only:
7
3
2
5
The result of addition of 4 number(s) is = 17

Process returned 0 (0x0)   execution time : 12.206 s
Press any key to continue.
```

Practice 9: A Program to Use if Statement

```
#include <iostream>

using namespace std;

int main()
{
int x = 50;
cout << "If Statement \n" << endl;

if(x > 35)
{
cout << "The values of x in above expected result" << endl;
}
cout << "You inserted " << x << endl;

return 0;
}
```

Output

```
If Statement

The values of x in above expected result
You inserted 50

Process returned 0 (0x0)   execution time : 0.164 s
Press any key to continue.
```

Practice 10: A Program to Use Nested if Statement

```
#include <iostream>

using namespace std;

int main()
{
int x, y;
x = 80;
y = 52;
cout << "Nested if Statement \n" << endl;

if(x == 80)
{
cout << "The first condition is met" << endl;
if(y == 52)
{
cout << "This is a NESTED if statement" << endl;
}
}

return 0;
}
```

Output

```
Nested if Statement

The first condition is met
This is a NESTED if statement

Process returned 0 (0x0)   execution time : 0.431 s
Press any key to continue.
```

Practice 11: A Program to Use if else Statement

```
#include <iostream>

using namespace std;

int main()
{
int x=67;

cout << "if else statement \n" << endl;

if(x == 69)
{
cout << "This is a correct if statement " << endl;
}
else
{
cout << "Incorrect password" << endl;
}

return 0;
}
```

Output

```
if else statement

Incorrect password

Process returned 0 (0x0)   execution time : 0.205 s
Press any key to continue.
```

Practice 12: A Program to Use else if Statement

```
#include <iostream>
using namespace std;

int main()
{
int grade;
cout << "else if statement \n" << endl;

cout << "Insert student's mark: " << endl;
cin >> grade;

if(grade > 100)
{
cout << "Incorrect grade" << endl;
}
else if(grade >= 80)
{
cout << "Your grade is A" << endl;
}
else if(grade >= 70)
{
cout << "Your grade is B" << endl;
}
else if(grade >= 50)
{
cout << "Your grade is C" << endl;
}
else if(grade >= 45)
{
cout << "Your grade is D" << endl;
}
else if(grade >= 35)
{
cout << "Your grade is D" << endl;
}
else {
cout << "You FAILED" << endl;
}
return 0;
}
```

Output

```
else if statement

Insert student's mark:
70
Your grade is B

Process returned 0 (0x0)   execution time : 3.424 s
Press any key to continue.
```

Practice 13: A Program to use Switch Statement

```
#include <iostream>
using namespace std;

int main()
{
char grade;
cout << "Switch Statement \n" << endl;

cout << "Please insert your grade from A-F only" << endl;
std::cin >> grade;

switch(grade)
{
case 'A':
cout << "Excellent Result" << endl;
break;

case 'B':
cout << "Very Good Result" << endl;
break;

case 'C':
cout << "Good Result" << endl;
break;

case 'D':
cout << "Well done Result" << endl;
break;

case 'E':
cout << "Passed Result" << endl;
break;

case 'F':
cout << "Failed Result" << endl;
break;

default:
cout << "Invalid grade" << endl;
4}
return 0;
}
```

Output

```
Switch Statement

Please insert your grade from A-F only
A
Excellent Result

Process returned 0 (0x0)   execution time : 2.479 s
Press any key to continue.
```

Practice 14: A Program for Nested Switch Statement

```
#include <iostream>

using namespace std;

int main()
{
int x = 87, y = 70;
cout << "Nested Switch case \n" << endl;

 switch(x)
{
case 87:
cout << "This is the first switch case" << endl;

switch(y)
{
case 70:
cout << "This is the second switch case" << endl;
}
}
return 0;
}
```

Output

```
Nested Switch case

This is the first switch case
This is the second switch case

Process returned 0 (0x0)   execution time : 0.133 s
Press any key to continue.
```

Practice 15: A Program to use While Loop

```
#include <iostream>

using namespace std;

int main()
{
int x = 10;
cout << "while loop \n" << endl;

while(x <= 20)
{
cout << "The value of x is : " << x << endl;
x = x+1;
}

return 0;
}
```

Output

```
while loop

The value of x is : 10
The value of x is : 11
The value of x is : 12
The value of x is : 13
The value of x is : 14
The value of x is : 15
The value of x is : 16
The value of x is : 17
The value of x is : 18
The value of x is : 19
The value of x is : 20

Process returned 0 (0x0)   execution time : 0.132 s
Press any key to continue.
```

Practice 16: A Program for do while loop

```
#include <iostream>

using namespace std;

int main()
{
int x = 1;
cout << "do while loop \n" << endl;

do
{

cout << "The value of x = " << x << endl;
x = x+1;

}while(x <= 20);

return 0;
}
```

Output

```
do while loop

The value of x = 1
The value of x = 2
The value of x = 3
The value of x = 4
The value of x = 5
The value of x = 6
The value of x = 7
The value of x = 8
The value of x = 9
The value of x = 10
The value of x = 11
The value of x = 12
The value of x = 13
The value of x = 14
The value of x = 15
The value of x = 16
The value of x = 17
The value of x = 18
The value of x = 19
The value of x = 20

Process returned 0 (0x0)   execution time : 0.206 s
Press any key to continue.
```

Practice 17: A Program to use for loop

```
#include <iostream>

using namespace std;

int main()
{
int x = 1;
cout << "for loop \n" << endl;

for(x=1; x<=20; x++)
{
cout << "The value of x = " << x <<endl;
}

return 0;
}
```

Output

```
for loop

The value of x = 1
The value of x = 2
The value of x = 3
The value of x = 4
The value of x = 5
The value of x = 6
The value of x = 7
The value of x = 8
The value of x = 9
The value of x = 10
The value of x = 11
The value of x = 12
The value of x = 13
The value of x = 14
The value of x = 15
The value of x = 16
The value of x = 17
The value of x = 18
The value of x = 19
The value of x = 20

Process returned 0 (0x0)   execution time : 0.120 s
Press any key to continue.
```

Practice 18: A Program to Print Prime Numbers

```
#include <iostream>
using namespace std;

int main()
{
int x, y;
cout << "Nested for loop \n" << endl;

for(x=2; x<=100; x++)
{
for(y=2; y<=(x/y); y++)
if(!(x%y))
{
break;
}
if(y>(x/y))
{
cout << x << " is a prime number" << endl;
}
}
return 0;
}
```

Output

```
Nested for loop

2 is a prime number
3 is a prime number
5 is a prime number
7 is a prime number
11 is a prime number
13 is a prime number
17 is a prime number
19 is a prime number
23 is a prime number
29 is a prime number
31 is a prime number
37 is a prime number
41 is a prime number
43 is a prime number
47 is a prime number
53 is a prime number
59 is a prime number
61 is a prime number
67 is a prime number
71 is a prime number
73 is a prime number
79 is a prime number
83 is a prime number
89 is a prime number
97 is a prime number
```

Practice 19: A Program for Area of a Triangle

```
#include <iostream>

using namespace std;

int main()
{
float base, height, area;
cout << "Area of Triangle \n" << endl;

cout << "Insert height of triangle: " << endl;
cin >> height;

cout << "Insert base of triangle: " << endl;
cin >> base;

area = (base/2)*height;

cout << "The area of the given triangle = " << area << endl;

return 0;
}
```

Output

```
Area of Triangle

Insert height of triangle:
18
Insert base of triangle:
4
The area of the given triangle = 36

Process returned 0 (0x0)   execution time : 8.050 s
Press any key to continue.
```

Practice 20: A Program to Find Even and Odd Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x;
cout << "Even/Odd numbers \n" << endl;

cout << "Insert any number: " << endl;
cin >> x;

if(x%2 == 0)
{
cout << x << " is an EVEN number" << endl;
}
else
{
cout << x << " is an ODD number" << endl;
}

return 0;
}
```

Output

```
Even/Odd numbers

Insert any number:
7
7 is an ODD number

Process returned 0 (0x0)   execution time : 3.522 s
Press any key to continue.
```

Practice 21: A Program to Add Digits from User

```
#include <iostream>

using namespace std;

int main()
{
int x, y, sum = 0;
cout << "Add a given digit \n" << endl;

cout << "Insert some numbers: " << endl;
cin >> x;

do{

y = x%10;

sum = sum+y;

x = x/10;

}while(x != 0);

cout << "The result of the added numbers = " << sum << endl;

return 0;
}
```

Output

```
Add a given digit

Insert some numbers:
546732
The result of the added numbers = 27

Process returned 0 (0x0)   execution time : 5.572 s
Press any key to continue.
```

Practice 22. How to Find Greatest of Three Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z;
cout << "Find greatest of three numbers with if statement \n" << endl;

cout << "Insert first number: \n" << endl;
cin >> x;

cout << "Insert second number: \n" << endl;
cin >> y;

cout << "Insert third number: \n" << endl;
cin >> z;

if (x >= y && x >= z)
{
cout << "\n" << x << " is the greatest number \n" << endl;
}

if(y >= x && y >= z)
{
cout << y << " is the greatest number \n" << endl;
}

if(z >= x && z >= y)
{
cout << z << " is the greatest number \n" << endl;
}
return 0;
}
```

Output

```
Find greatest of three numbers with if statement

Insert first number:

67
Insert second number:

84
Insert third number:

13
84 is the greatest number


Process returned 0 (0x0)   execution time : 9.407 s
Press any key to continue.
```

Practice 23: A Program to Swap Numbers using Variables

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z;
cout << "How to swap Numbers using three variables \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

cout << "Numbers before swapping are : " << x << " and " << y <<endl;

z = x;
x = y;
y = z;

cout << "Numbers after swapping are : " << x << " and " << y <<endl;

return 0;
}
```

Output

```
How to swap Numbers using three variables

Insert first number:
14
Insert second number:
30
Numbers before swapping are : 14 and 30
Numbers after swapping are : 30 and 14

Process returned 0 (0x0)   execution time : 8.000 s
Press any key to continue.
```

Practice 24: A Program to Swap Numbers using 2 Variables

```
#include <iostream>

using namespace std;

int main()
{
int x, y;
cout << "Swap Numbers using two variables \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

cout << "Numbers before swapping are: " << x << " and " << y << endl;

x = x + y;
y = x - y;
x = x - y;

cout << "Numbers after swapping are: " << x << " and " << y << endl;

return 0;
}
```

Output

```
Swap Numbers using two variables

Insert first number:
5
Insert second number:
9
Numbers before swapping are: 5 and 9
Numbers after swapping are: 9 and 5

Process returned 0 (0x0)   execution time : 147.772 s
Press any key to continue.
```

Practice 25: A Program to Calculate the Percentage

```
#include <iostream>

using namespace std;

int main()
{
int sum, s1, s2, s3, s4, s5, s6, s7, total = 800;
float per;

cout << "Calculate Percentage \n" << endl;

cout << "Insert the marks of 7 subjects: \n" << endl;
cin >> s1 >> s2 >> s3 >> s4 >> s5 >> s6 >>s7;

sum = s1+s2+s3+s4+s5+s6+s7;

cout << "The sum of 7 subjects = " << sum << endl;

per = (sum * 100)/total;

cout << "\n Percentage = " << per << endl;

return 0;
}
```

Output

```
Calculate Percentage

Insert the marks of 7 subjects:

78
45
90
56
30
45
67
The sum of 7 subjects = 411

 Percentage = 51

Process returned 0 (0x0)   execution time : 33.198 s
Press any key to continue.
```

Practice 26: A Program to Calculate Gross Salary

```
#include <iostream>

using namespace std;

int main()
{
int basic, x,y,gro_sa;

cout << "Gross Salary \n" << endl;

cout << "Insert your basic salary: " << endl;
cin >> basic;

x = (10*basic)/100;
y = (12*basic)/100;

gro_sa = basic + x + y;

cout << "Your Gross salary = " << gro_sa << endl;


return 0;
}
```

Output

```
Gross Salary

Insert your basic salary:
10000
Your Gross salary = 12200

Process returned 0 (0x0)   execution time : 6.415 s
Press any key to continue.
```

Practice 27: A Program to Calculate Simple Interest

```
#include <iostream>

using namespace std;

int main()
{
int principal_amount;
float rate, period, simple_interest;

cout << "Simple Interest \n" << endl;

cout << "Insert Principal Amount: " << endl;
cin >> principal_amount;

cout << "Insert Rate: " << endl;
cin >> rate;

cout << "Insert the period: " << endl;
cin >> period;

simple_interest = (principal_amount * rate * period)/100;

cout << "\n The result of the simple interest = " << simple_interest
<< endl;

return 0;
}
```

Output

```
Simple Interest

Insert Principal Amount:
100
Insert Rate:
50
Insert the period:
7

 The result of the simple interest = 350

Process returned 0 (0x0)   execution time : 13.582 s
Press any key to continue.
```

Practice 28: A Program to Check a Leap Year

```
#include <iostream>

using namespace std;

int main()
{
int yr;
cout << "A program to detect a leap year \n" << endl;

cout << "Insert any year of your choice: " << endl;
cin >> yr;

if(yr%4 == 0)
{
cout << yr << " is a LEAP year!" << endl;
}
else
{
cout << yr << " is NOT a leap year!" << endl;
}

return 0;
}
```

Output

```
A program to detect a leap year

Insert any year of your choice:
2030
2030 is NOT a leap year!

Process returned 0 (0x0)   execution time : 3.870 s
Press any key to continue.
```

Practice 29: A Program to Get HCF using Recursive Function

```
#include <iostream>

using namespace std;

int HCF(int x, int y);

int main()
{
int x, y;

cout << "Calculate HCF using recursive function \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

cout << "The HCH of " << x << " and " << y << " is " << HCF(x,y);

return 0;
}

int HCF(int x, int y)
{
if(y != 0)
return HCF(y, x%y);
else
return x;
}
```

Output

```
Calculate HCF using recursive function

Insert first number:
350
Insert second number:
90
The HCH of 350 and 90 is 10
Process returned 0 (0x0)   execution time : 11.144 s
Press any key to continue.
```

Practice 30: A Program to Calculate the LCM

```
#include <iostream>

using namespace std;

int main()
{
int x, y, maxi;
cout << "Calculate LCM \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

maxi = (x>y)?x:y;

while(true)
{
if(maxi%x == 0 && maxi%y == 0)
{
cout << "The LCM of " << x << " and " << y << " = " << maxi << endl;
break;
}
else
{
maxi++;
}
}

return 0;
}
```

Output

```
Calculate LCM

Insert first number:
100
Insert second number:
45
The LCM of 100 and 45 = 900

Process returned 0 (0x0)   execution time : 12.289 s
Press any key to continue.
```

Practice 31: A Program to Calculate LCM and HCF

```
#include <iostream>
using namespace std;

long HCFLCM(long x, long y);

int main()
{
long x, y, HCF, LCM;
cout << "Calculate LCM and HCF \n" << endl;
cout << "Insert first number: " << endl;
cin >> x;
cout << "Insert second number: " << endl;
cin >> y;

HCF = HCFLCM(x,y);
LCM = (x*y)/HCF;

cout << "The HCF of " << x << " and " << y << " is " << HCF << endl;
cout << "\n The LCM of " << x << " and " << y << " is " << LCM << endl;
return 0;
}

long HCFLCM(long x, long y)
{
if(x == 0)
{
return y;
}
do{
if(x>y)
{
x = x - y;
}
else
{
y = y - x;
}
}while( y != 0);
return x;
}
```

Output

```
Calculate LCM and HCF

Insert first number:
100
Insert second number:
45
The HCF of 100 and 45 is 5

 The LCM of 100 and 45 is 900

Process returned 0 (0x0)   execution time : 3.932 s
Press any key to continue.
```

Practice 32: A Program to Get Factorial of nCr and nPr

```
#include <iostream>
using namespace std;

long fact_ncr(int n, int r);

long fact_npr(int n, int r);

long factorial(int n);

int main()
{
int n,r;
long ncr, npr;
cout << "Factorial of nCr and nPr \n" << endl;

cout << "Insert the value of n: " << endl;
cin >> n;

cout << "Insert the value of r: " << endl;
cin >> r;

ncr = fact_ncr(n,r);
npr = fact_npr(n,r);

cout << n << "C" << r << " = " << ncr << endl;
cout << n << "P" << r << " = " << npr << endl;

return 0;
}

long fact_ncr(int n, int r)
{
long result;
result = factorial(n)/(factorial(r)*factorial(n-r)); //n!/(r!(n-r)!)

return result;
}

long fact_npr(int n, int r)
{
long result;
result = factorial(n)/factorial(n-r); //n!/(n-r)!
return result;
}

long factorial(int n)
{
int x;
long result = 1;
for(x=1; x<=n; x++)
result = result*x;

return(result);
}
```

Output

```
Factorial of nCr and nPr

Insert the value of n:
16
Insert the value of r:
20
16C20 = 0
16P20 = 2004189184

Process returned 0 (0x0)   execution time : 8.157 s
Press any key to continue.
```

Practice 33: A Program to Reverse Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, rev = 0, y;
cout << "How to Reverse Numbers \n" << endl;

cout << "Insert some numbers: " << endl;
cin >> x;

while(x != 0)
{
y = x%10;
rev = rev*10 + y;
x = x/10;
}

cout << "The reversed of numbers = " << rev << endl;

return 0;
}
```

Output

```
How to Reverse Numbers

Insert some numbers:
123456789
The reversed of numbers = 987654321

Process returned 0 (0x0)   execution time : 6.062 s
Press any key to continue.
```

Practice 34: A Program to Reverse Arrays of Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, y, rev[100];
cout << "Reverse of Array of Numbers \n" << endl;

cout << "Insert range of number: " << endl;
cin >> x;

cout << "Insert only " << x << " numbers: " << endl;
for(y=0; y<x; y++)
{
cin >> rev[y];
}

cout << "The reversed numbers are: \n" << endl;
for(y=x-1; y>=0; y--)
{
cout << rev[y] << endl;
}
return 0;
}
```

Output

```
Reverse of Array of Numbers

Insert range of number:
5
Insert only 5 numbers:
9
4
2
8
6
The reversed numbers are:

6
8
2
4
9

Process returned 0 (0x0)   execution time : 34.981 s
Press any key to continue.
```

Practice 35: A Program to Check for a Palindrome

```
#include <iostream>

using namespace std;

int main()
{
int x, y, pal, rev=0;
cout << "Checking a Palindrome \n" << endl;

cout << "Insert only positive numbers: " << endl;
cin >> x;

y = x;

while(x != 0)
{
pal = x%10;

rev = (rev*10) + pal;

x = x/10;
}

if(y == rev)
{
cout << "This number is a PALINDROME!" << endl;
}
else
{
cout << "This number is NOT a palindrome" << endl;
}

return 0;
}
```

Output

```
Checking a Palindrome

Insert only positive numbers:
3456543
This number is a PALINDROME!

Process returned 0 (0x0)   execution time : 7.254 s
Press any key to continue.
```

Practice 36: A Program to Generate Prime Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, cnt=0;
cout << "Check Prime Numbers \n" << endl;

cout << "Insert any number to check for Prime number: " << endl;
cin >> x;

for(int y=1; y<=x; y++)
{
if(x%y == 0)
{
cnt++;
}
}

if(cnt == 2)
{
cout << "This number is a PRIME number!" << endl;
}
else
{
cout << "This number is NOT a prime number!" << endl;
}

return 0;
}
```

Output

```
Check Prime Numbers

Insert any number to check for Prime number:
23
This number is a PRIME number!

Process returned 0 (0x0)   execution time : 4.754 s
Press any key to continue.
```

Practice 37: A Program to Print List of Prime Numbers

```
#include <iostream>
using namespace std;
int list_prime(int n);

int main()
{
int x, y, z;
cout << "List of Prime numbers up to a given number \n" << endl;
cout << "Choose the range of prime numbers: " << endl;
cin >> x;

for(y=0; y<x; y++)
{
z = list_prime(y);

if(z == 1)
{
cout << y << endl;
}
}
return 0;
}

int list_prime(int n)
{
int x;
for(x=2; x<=n-1; x++)
{
if(n%x == 0)
return 0;
}
if(x == n)
{
return 1;
}
}
```

Output

```
List of Prime numbers up to a given number

Choose the range of prime numbers:
7
2
3
5

Process returned 0 (0x0)   execution time : 4.430 s
Press any key to continue.
```

Practice 38: A Program to Check for Armstrong Numbers

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z=0, temp, rem;
cout << "Armstrong Numbers \n" << endl;

cout << "Insert any number to check for Armstrong: " << endl;
cin >> x;

temp = x;

while(temp != 0)
{
rem = temp%10;
z = z + rem*rem*rem;
temp = temp/10;
}

if(x == z)
{
cout << "This number is an ARMSTRONG number!" << endl;
}
else
{
cout << "This number is NOT an Armstrong number!" << endl;
}

return 0;
}
```

Output

```
Armstrong Numbers

Insert any number to check for Armstrong:
153
This number is an ARMSTRONG number!

Process returned 0 (0x0)   execution time : 6.985 s
Press any key to continue.
```

Practice 39: A Program to Generate Armstrong Numbers

```
#include <iostream>
using namespace std;

int main()
{
int rem, x;
long y=0, z=0, temp;
cout << "Generate Armstrong Numbers \n" << endl;

cout << "Insert range of Armstrong numbers: " << endl;
cin >> y;

cout << "The list of Armstrong numbers up to " << y << " are " << endl;
for(x=1; x<=y; x++)
{
temp = x;
while(temp != 0)
{
rem = temp%10;
z = z+rem*rem*rem;
temp = temp/10;
}
if(x == z)
{
cout << x << endl;
}
z = 0;
}

return 0;
}
```

Output

```
Generate Armstrong Numbers

Insert range of Armstrong numbers:
10000
The list of Armstrong numbers up to 10000 are
1
153
370
371
407

Process returned 0 (0x0)   execution time : 8.955 s
Press any key to continue.
```

Practice 40: A Program to Calculate Factorial

```
#include <iostream>

using namespace std;

int main()
{
int x, fact = 1;
cout << "Factorial \n" << endl;

cout << "Insert any number to find the factorial: " << endl;
cin >> x;

for(int y = 1; y<=x; y++)
{
fact = fact*y;
}

cout << "The factorial of the given number = " << fact << endl;

return 0;
}
```

Output

```
Factorial

Insert any number to find the factorial:
5
The factorial of the given number = 120

Process returned 0 (0x0)   execution time : 4.231 s
Press any key to continue.
```

Practice 41: A Program to Calculate Factorial using Recursive Function

```
#include <iostream>
using namespace std;

int fact(int n);

int main()
{
int x;
cout << "Factorial with Recursive function \n" << endl;

cout << "Insert any number to find the factorial: " << endl;
cin >> x;

cout << "The result of the factorial is " << fact(x) << endl;

return 0;
}

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

Output

```
Factorial with Recursive function

Insert any number to find the factorial:
7
The result of the factorial is 5040

Process returned 0 (0x0)   execution time : 3.179 s
Press any key to continue.
```

Practice 42: A Program to Create a Fibonacci Series

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z=0, n=1;
cout << "Fibonacci Series \n" << endl;

cout << "Choose number of terms in series: " << endl;
cin >> x;

cout << "The result of the Fibonacci series is : " << endl;

for(int a=0; z<x; a++)
{
if(a<=1)
{
y = a;
}
else
{
y = z+n;
z=n;
n=y;
}
cout << y << endl;
}

return 0;
}
```

Output

```
Fibonacci Series

Choose number of terms in series:
5
The result of the Fibonacci series is :
0
1
1
2
3
5
8

Process returned 0 (0x0)   execution time : 18.196 s
Press any key to continue.
```

Practice 43: A Program for Fibonacci Series using Recursive Function

```
#include <iostream>
using namespace std;

int Fibonacci(int x);

int main()
{
int x = 0, y;
cout << "Fibonacci Series with recursive function \n" << endl;
cout << "Insert number of series: " << endl;
cin >> y;
cout << "The result of Fibonacci series is: " << endl;
for(int z=1; z<=y; z++)
{
cout << Fibonacci(x) << endl;
x++;
}
return 0;
}

int Fibonacci(int x)
{
if(x == 0)
{
return 0;
}
else if(x == 1)
{
return 1;
}
else
{
return(Fibonacci(x-1) + Fibonacci(x-2));
}
}
```

Output

```
Fibonacci Series with recursive function

Insert number of series:
8
The result of Fibonacci series is:
0
1
1
2
3
5
8
13

Process returned 0 (0x0)   execution time : 3.839 s
Press any key to continue.
```

Practice 44: A Program for Mathematical Functions

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
int x = 100, y = -23;
float a = 56.8, b = 100;
cout << "Mathematical Functions \n" << endl;

cout << "sin(x) = " << sin(x) << endl;
cout << "log(x) = " << log(x) << endl;
cout << "abs(y) = " << abs(y) << endl;
cout << "floor(a) = " << floor(a) << endl;
cout << "sqrt(b) = " << sqrt(b) << endl;
cout << "pow(b,2) = " << pow(b,2) << endl;

return 0;
}
```

Output

```
Mathematical Functions

sin(x) = -0.506366
log(x) = 4.60517
abs(y) = 23
floor(a) = 56
sqrt(b) = 10
pow(b,2) = 10000

Process returned 0 (0x0)   execution time : 0.652 s
Press any key to continue.
```

Practice 45: A Program to Generate Random Numbers

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main()
{
int x, y, maxi, ran;
cout << "Random Number Generator \n" << endl;

cout << "Choose range of random numbers to generate: " << endl;
cin >> x;

cout << "Choose maximum value of random numbers: " << endl;
cin >> maxi;

cout << "The result of the random numbers are: " << endl;

for(y=0; y<=x; y++)
{
ran = rand()%maxi;
cout << ran << endl;
}

return 0;
}
```

Output

```
Random Number Generator

Choose range of random numbers to generate:
8
Choose maximum value of random numbers:
1000
The result of the random numbers are:
41
467
334
500
169
724
478
358
962

Process returned 0 (0x0)   execution time : 13.726 s
Press any key to continue.
```

Practice 46: A Program to Convert from Binary to Hexadecimal

```
#include <iostream>

using namespace std;

int main()
{
long int bin, hex = 0, x=1, y, z;
cout << "Binary to Hexadecimal \n" << endl;

cout << "Insert binary numbers only: " << endl;
cin >> bin;

z = bin;

while(bin != 0)
{
y = bin%10;
hex = hex + y*x;
x = x*2;
bin = bin/10;
}

cout << "The Hexadecimal equivalent of " << z << " is " << hex << endl;

return 0;
}
```

Output

```
Binary to Hexadecimal

Insert binary numbers only:
1111
The Hexadecimal equivalent of 1111 is 15

Process returned 0 (0x0)   execution time : 3.343 s
Press any key to continue.
```

Practice 47: A Program to Convert from Binary to Octal

```
#include <iostream>

using namespace std;

int main()
{
long int bin, oct = 0, x = 1, y, z;
cout << "Binary to Octal \n" << endl;

cout << "Insert binary numbers only: " << endl;
cin >> bin;

z = bin;

while(bin != 0)
{
y = bin%10;
oct = oct + y*x;
x = x*2;
bin = bin/10;
}

cout << "The Octal equivalent of " << z << " is " << oct << endl;

return 0;
}
```

Output

```
Binary to Octal

Insert binary numbers only:
1101
The Octal equivalent of 1101 is 13

Process returned 0 (0x0)   execution time : 10.886 s
Press any key to continue.
```

Practice 48: A Program to Convert from Binary to Decimal

```
#include <iostream>

using namespace std;

int main()
{
int bin, dec = 0, x = 1, y, z;
cout << "Binary to Decimal \n" << endl;

cout << "Insert binary numbers only: " << endl;
cin >> bin;

z = bin;

while(bin > 0)
{
y = bin%10;
dec = dec + y*x;
bin = bin/10;
bin = bin/10;
x = x*2;
}

cout << "The Decimal equivalent of " << z << " is " << dec << endl;

return 0;
}
```

Output

```
Binary to Decimal

Insert binary numbers only:
1110
The Decimal equivalent of 1110 is 2

Process returned 0 (0x0)   execution time : 8.838 s
Press any key to continue.
```

Practice 49: A Program to Convert from Decimal to Octal

```
#include <iostream>

using namespace std;

int main()
{
long x, dec, y, ba = 1, oct=0;
cout << "Decimal to Octal \n" << endl;

cout << "Insert a Decimal Number: " << endl;
cin >> x;

dec = x;

while(x > 0)
{
y = x%8;
oct = oct + y*ba;
x = x/8;
ba = ba*10;
}

cout << "The Octal equivalent of " << dec << " is " << oct << endl;

return 0;
}
```

Output

```
Decimal to Octal

Insert a Decimal Number:
10
The Octal equivalent of 10 is 12

Process returned 0 (0x0)   execution time : 2.518 s
Press any key to continue.
```

Practice 50: A Program to Convert from Decimal to Binary

```
#include <iostream>

using namespace std;

int main()
{
long x, dec, y, ba = 1, bin = 0;
cout << "Decimal to Binary \n" << endl;

cout << "Insert Decimal Number: " << endl;
cin >> x;

dec = x;

while(x > 0)
{
y = x%2;
bin = bin + y*ba;
x = x/2;
ba = ba*10;
}

cout << "The Binary equivalent of " << dec << " is " << bin << endl;

return 0;
}
```

Output

```
Decimal to Binary

Insert Decimal Number:
9
The Binary equivalent of 9 is 1001

Process returned 0 (0x0)   execution time : 87.539 s
Press any key to continue.
```

Practice 51: A Program to Convert from Decimal to Bitwise

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, a;
cout << "Decimal to Bitwise \n" << endl;

cout << "Insert a Decimal Number: " << endl;
cin >> x;

a = x;

cout << "The Binary Bitwise equivalent of " << a << endl;
for(y=31; y>=0; y--)
{
z = x >> y;
if(y & 1)
{
cout << "1" << endl;
}
else
{
cout << "0" << endl;
}
}

return 0;
}
```

Output

```
Decimal to Bitwise

Insert a Decimal Number:
4
The Binary Bitwise equivalent of 4
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
1
0
```

Practice 52: A Program to Find First and Last Number in an Array

```
#include <iostream>

using namespace std;

int main()
{
int x, n, Arr[100];
cout << "Detect first and last values in an Array \n" << endl;

cout << "Choose range of Array element: " << endl;
cin >> n;

cout << "Insert only " << n << " numbers" << endl;
for(x=0; x<n; ++x)
{
cin >> Arr[x];
}

cout << "The first number is " << Arr[0] << " and " << " the last number is "
<< Arr[n-1];

return 0;
}
```

Output

```
Detect first and last values in an Array

Choose range of Array element:
4
Insert only 4 numbers
2
3
4
5
The first number is 2 and  the last number is 5
Process returned 0 (0x0)   execution time : 9.036 s
Press any key to continue.
```

Practice 53: A Program to Pass Arrays to Functions

```
#include <iostream>

using namespace std;

double getAvg(int arr[], int size);

int main()
{
int amt[8] = {87,50,30,90,87,45,100,212};
double avg;

cout << "Passing Array to Function \n" << endl;

avg = getAvg(amt,8);

cout << "The average of the amount = " << avg << endl;

return 0;
}

double getAvg(int arr[], int size)
{
int x, y=0;
double avg;

for(x = 0; x<size; ++x)
{
y += arr[x];
}

avg = double(y)/size;

return avg;
}
```

Output

```
Passing Array to Function

The average of the amount = 87.625

Process returned 0 (0x0)   execution time : 0.387 s
Press any key to continue.
```

Practice 54: A Program to Find Maximum Element in Array

```
#include <iostream>

using namespace std;

int main()
{
int amt[200], maxi, len,pos = 1;

cout << "Maximum Element in an Array \n" << endl;

cout << "Choose range of array elements: " << endl;
cin >> len;

cout << " Insert "<< len <<" Elements of the array only" << endl;
for(int x=0; x<len; x++)
{
cin >> amt[x];
}

maxi = amt[0];
for(int x=1; x<len; x++)
{
if(amt[x] > maxi)
{
maxi = amt[x];
pos = x+1;
}
}

cout << "The maximum element is at position "<< pos << " and it's value is "
<< maxi << endl;

return 0;
}
```

Output

```
Maximum Element in an Array

Choose range of array elements:
3
 Insert 3 Elements of the array only
4
5
6
The maximum element is at position 3 and it's value is 6

Process returned 0 (0x0)   execution time : 10.497 s
Press any key to continue.
```

Practice 55: A Program to Find the Minimum Element in an Array

```
#include <iostream>

using namespace std;

int main()
{
int amt[200], mini, len, pos = 1;
cout << "Minimum Element in Array \n" << endl;

cout << "Choose the range of array element: " << endl;
cin >> len;

cout << "Insert " << len << " array elements only: " << endl;
for(int x=0; x<len; x++)
{
cin >> amt[x];
}

mini = amt[0];

for(int x=1; x<len; x++)
{
if(amt[x] < mini)
{
mini = amt[x];
pos = x+1;
}
}

cout << "The minimum element is at position "<< pos << " and its value is "
<< mini << endl;

return 0;
}
```

Output

```
Minimum Element in Array

Choose the range of array element:
4
Insert 4 array elements only:
45
65
32
12
The minimum element is at position 4 and its value is 12

Process returned 0 (0x0)   execution time : 6.784 s
Press any key to continue.
```

Practice 56: A Program to Reverse Array Elements

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, amt1[200], amt2[200];
cout << "Reverse Array Elements \n" << endl;

cout << "Choose range of elements: " << endl;
cin >> x;

cout << "Insert " << x << "array elements only: " << endl;
for(y=0; y<x; y++)
{
cin >> amt1[y];
}

for(y=y-1, z=0; y>=0; y--, z++)
{
amt2[z] = amt1[y];
}

for(y=0; y<x; y++)
{
amt1[y] = amt2[y];
}

cout << "The reversed array element is: " << endl;
for(y=0; y<x; y++)
{
cout << amt1[y] << endl;
}

return 0;
}
```

Output

```
Reverse Array Elements

Choose range of elements:
4
Insert 4array elements only:
78
40
35
2
The reversed array element is:
2
35
40
78

Process returned 0 (0x0)   execution time : 11.970 s
Press any key to continue.
```

Practice 57: A Program to Insert a New Element into an Array

```
#include <iostream>
using namespace std;

int main()
{
int amt[200], pos, x, y, num;
cout << "Insert Element in an Array \n" << endl;

cout << "Choose range of array elements: " << endl;
cin >> x;

cout << "Insert " << x << " array elements only" << endl;
for(y=0; y<x; y++)
{
cin >> amt[y];
}
cout << "Choose a position to insert new element: " << endl;
cin >> pos;

cout << "Insert the new element: " << endl;
cin >> num;

for(y=x-1; y>=pos-1; y--)
{
amt[y+1] = amt[y];
}
amt[pos - 1] = num;

cout << "The new array element after insertion is: " << endl;
for(y=0; y<=x; y++)
{
cout << amt[y] << endl;
}
return 0;
}
```

Output

```
Insert Element in an Array

Choose range of array elements:
4
Insert 4 array elements only
7
9
5
8
Choose a position to insert new element:
2
Insert the new element:
100
The new array element after insertion is:
7
100
9
5
8

Process returned 0 (0x0)   execution time : 103.379 s
Press any key to continue.
```

Practice 58: A Program to Delete an Element in Array

```
#include <iostream>
using namespace std;

int main()
{
int x, y, amt[200], pos;
cout << "Delete an Element from Array \n " << endl;
cout << "Choose range of array element" << endl;
cin >> x;
cout << "Insert " << x << " array elements only" << endl;
for(y=0; y<x; y++)
{
cin >> amt[y];
}

cout << "Choose the position of the element you want to delete: " << endl;
cin >> pos;

if(pos >= x+1)
{
cout << "The operation is not possible..." << endl;
}
else
{
for(y=pos-1; y<x-1; y++)
{
amt[y] = amt[y+1];
}
}
cout << "The result of the array after deletion is: " << endl;
for(y=0; y<x-1; y++)
{
cout << amt[y] << endl;
}
return 0;
}
```

Output

```
Delete an Element from Array

Choose range of array element
4
Insert 4 array elements only
56
43
78
60
Choose the position of the element you want to delete:
2
The result of the array after deletion is:
56
78
60

Process returned 0 (0x0)   execution time : 10.794 s
Press any key to continue.
```

Practice 59: A Program to Merge 2 Arrays

```
#include <iostream>

using namespace std;

void merged_amt(int arr1[], int x, int arr2[], int y, int total[]);

int main()
{
int x, y, a, amt1[200], amt2[200], sum[200];
cout << "A program to Merge Arrays \n" << endl;

cout << "Choose range of first array element: " << endl;
cin >> x;

cout << "Insert first array of " << x << " elements only!" << endl;
for(a=0; a<x; a++)
{
cin >> amt1[a];
}

cout << "Choose range of second array elements: " << endl;
cin >> y;

cout << "Insert second array of " << y << " elements only!" << endl;
for(a=0; a<y; a++)
{
cin >> amt2[a];
}

merged_amt(amt1, x, amt2, y, sum);

cout << "The result of the merged array is: " << endl;
for(a=0; a<x+y; a++)
{
cout << sum[a] << endl;
}

return 0;
}

void merged_amt(int arr1[], int x, int arr2[], int y, int total[])
{
int a, b=0, c=0;

for(a=0; a< x+y;)
{
if(b<x && c<y)
{
if(arr1[b] < arr2[c])
{
total[a] = arr1[b];
b++;
}
else
```

```
{
total[a] = arr2[c];
c++;
}
a++;
}
else if(b == x)
{
for(;a<x+y;)
{
total[a] = arr2[c];
c++;
a++;
}
}
else
{
for(;a<x+y;)
{
total[a] = arr1[b];
b++;
a++;
}
}
}
}
```

Output

```
A program to Merge Arrays

Choose range of first array element:
4
Insert first array of 4 elements only!
56
87
32
12
Choose range of second array elements:
3
Insert second array of 3 elements only!
78
90
14
The result of the merged array is:
56
78
87
32
12
90
14

Process returned 0 (0x0)   execution time : 24.868 s
Press any key to continue.
```

Practice 60: A Program to Add Two Matrices

```
#include <iostream>

using namespace std;

int main()
{
int x, y, r, c, mat1[100][100], mat2[100][100], sum[200][200];
cout << "Addition of two Matrices \n" << endl;

cout << "Insert number of rows: " << endl;
cin >> x;

cout << "Insert number of columns: " << endl;
cin >> y;

cout << "Insert elements of first matrix: " << endl;
for(r=0; r<x; r++)
{
for(c=0; c<y; c++)
{
cin >> mat1[r][c];
}
}

cout << "Insert elements of second matrix: " << endl;
for(r=0; r<x; r++)
{
for(c=0; c<y; c++)
{
cin >> mat2[r][c];
}
}

for(r=0; r<x; r++)
{
for(c=0; c<y; c++)
{
sum[r][c] = mat1[r][c] + mat2[r][c];
}
}

cout << "The result of matrix is: " << endl;
for(r=0; r<x; r++)
{
for(c=0; c<y; c++)
{
cout << sum[r][c] << endl;
}
cout << "\n" << endl;
}

return 0;
}
```

Output

```
Addition of two Matrices

Insert number of rows:
3
Insert number of columns:
3
Insert elements of first matrix:
34
23
12
56
78
90
21
34
54
Insert elements of second matrix:
3
4
5
6
7
8
9
1
3
The result of matrix is:
37
27
17


62
85
98


30
35
57
```

Practice 61: A Program to Subtract Two Matrices

```
#include <iostream>

using namespace std;

int main()
{
int x, y, row, col, mat1[200][200], mat2[200][200], sub[200][200];
cout << "Subtraction of Matrices \n" << endl;

cout << "Insert the number of rows: " << endl;
cin >> x;

cout << "Insert the number of column: " << endl;
cin >> y;

cout << "Insert elements of first array: " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
cin >> mat1[row][col];
}
}

cout << "Insert elements of second array: " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
cin >> mat2[row][col];
}
}

for(row=0;row<x; row++)
{
for(col=0; col<y; col++)
{
sub[row][col] = mat1[row][col] - mat2[row][col];
}
}

cout << "The result of the subtraction of the two matrices is: " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
cout << sub[row][col] << endl;
}
cout << "\n" << endl;
}

return 0;
}
```

Output

```
Subtraction of Matrices

Insert the number of rows:
3
Insert the number of column:
3
Insert elements of first array:
45
23
21
4
67
78
98
45
32
Insert elements of second array:
56
67
32
45
2
8
9
5
4
The result of the subtraction of the two matrices is:
-11
-44
-11


-41
65
70


89
40
28



Process returned 0 (0x0)   execution time : 31.239 s
```

Practice 62: A Program to Transpose a Matrix

```
#include <iostream>

using namespace std;

int main()
{
int x, y, row, col, mat[100][100], trans[100][100];
cout << "Transpose Matrix \n" << endl;

cout << "Insert numbers of rows: " << endl;
cin >> x;

cout << "Insert numbers of column: " << endl;
cin >> y;

cout << "Insert elements of the matrix : " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
cin >> mat[row][col];
}
}

for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
trans[row][col] = mat[row][col];
}
}

cout << "The result of the transpose is: " << endl;
for(row=0; row<y; row++)
{
for(col=0; col<x; col++)
{
cout << trans[row][col] << endl;
}
cout << "\n" << endl;
}

return 0;
}
```

Output

```
Transpose Matrix

Insert numbers of rows:
3
Insert numbers of column:
3
Insert elements of the matrix :
6
5
4
7
8
9
3
2
6
The result of the transpose is:
6
5
4


7
8
9


3
2
6


Process returned 0 (0x0)   execution time : 14.692 s
Press any key to continue.
```

Practice 63: A Program to Multiply 2 Matrices

```
#include <iostream>

using namespace std;

int main()
{
int x, y, row, col, a, b, z, sum=0, mat1[20][20], mat2[20][20], mul[20][20];
cout << "Multiplication of two Matrices \n" << endl;

cout << "Insert number of rows of first matrix: " << endl;
cin >> x;

cout << "Insert number of column of first matrix: " << endl;
cin >> y;

cout << "Insert elements of the first matrix: " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<y; col++)
{
cin >> mat1[row][col];
}
}

cout << "Insert number of rows of second matrix: " << endl;
cin >> a;

cout << "Insert number of column second matrix: " << endl;
cin >> b;

if(y != a)
{
cout << "The order of the matrices cannot be multiplied" << endl;
}
else
{
cout << "Insert elements of the second matrix: " << endl;
for(row=0; row<a; row++)
{
for(col=0; col<b; col++)
{
cin >> mat2[row][col];
}
}

for(row=0; row<x; row++)
{
for(col=0; col<b; col++)
{
for(z=0; z<a; z++)
{
sum = sum+mat1[row][z]*mat2[z][col];
}
mul[row][col] = sum;
```

```
sum = 0;
}
}
}

cout << "Multiplication of matrices is : " << endl;
for(row=0; row<x; row++)
{
for(col=0; col<b; col++)
{
cout << mul[row][col] << endl;
}
cout << "\n" << endl;
}

return 0;
}
```

Output

```
Multiplication of two Matrices

Insert number of rows of first matrix:
2
Insert number of column of first matrix:
2
Insert elements of the first matrix:
56
78
4
2
Insert number of rows of second matrix:
2
Insert number of column second matrix:
2
Insert elements of the second matrix:
5
23
8
9
Multiplication of matrices is :
904
1990


36
110



Process returned 0 (0x0)   execution time : 20.125 s
Press any key to continue.
```

Practice 64: A Program for Simple String Format

```
#include <iostream>

using namespace std;

int main()
{
char STR[100] = {'H','e','l','l','o',' ', 'W','o','r','l','d'};
cout << "Simple String Format in C++ \n" << endl;

cout << "The string output is: " << STR << endl;

return 0;
}
```

Output

```
Simple String Format in C++

The string output is: Hello World

Process returned 0 (0x0)   execution time : 0.334 s
Press any key to continue.
```

Practice 65: A Program to Check for Length of String using for Loop

```
#include <iostream>

using namespace std;

int main()
{
int x, cnt=0;
char STR[20];
cout << "String Length with for loop \n" << endl;

cout << "Insert strings of any character: " << endl;
cin >> STR;


for(x=0; STR[x]!='\0'; x++)
{

cnt++;
}

cout << "The length of the given string is: " << cnt << endl;

return 0;
}
```

Output

```
String Length with for loop

Insert strings of any character:
IfeanyiChukwu
The length of the given string is: 13

Process returned 0 (0x0)   execution time : 6.370 s
Press any key to continue.
```

Practice 66: A Program to Check for Length of String using Function

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
char STR[100];
int len;
cout << "Find String length using function \n" << endl;

cout << "Insert string of any length: " << endl;
cin >> STR;

len = strlen(STR);

cout << "The length of the string is : " << len << endl;

return 0;
}
```

Output

```
Find String length using function

Insert string of any length:
IfeanyiChukwu
The length of the string is : 13

Process returned 0 (0x0)   execution time : 11.957 s
Press any key to continue.
```

Practice 67: A Program to Compare Strings using for loop

```
#include <iostream>

using namespace std;

int main()
{
char str1[100], str2[200], x = 0, y = 0, flag=0;
cout << "Comparing two Strings using for loop \n" << endl;

cout << "Insert first string: " << endl;
cin >> str1;

cout << "Insert second string: " << endl;
cin >> str2;

while(str1[x] != '\0')
{
x++;
}

while(str2[y] != '\0')
{
y++;
}

if(x != y)
{
flag = 0;
}
else
{
for(x=0,y=0; str1[x]!='\0',str2[y]!='\0'; x++,y++)
{
if(str1[x] == str2[y])
{
flag = 1;
}
}
}
if(flag==0)
{
cout << "Strings are NOT equal!" << endl;
}
else{
cout << "Strings are EQUAL!" << endl;
}

return 0;
}
```

Output

```
Comparing two Strings using for loop

Insert first string:
Ejike
Insert second string:
IfeanyiChukwu
Strings are NOT equal!

Process returned 0 (0x0)   execution time : 10.633 s
Press any key to continue.
```

Practice 68: A Program to Compare two Strings using Function

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str1[200], str2[100];
cout << "Compare Strings using Strcmp() function \n" << endl;

cout << "Insert first string: " << endl;
cin >> str1;

cout << "Insert second string: " << endl;
cin >> str2;


if(strcmp(str1,str2) == 0)
{
cout << "The strings are EQUAL" << endl;
}
else
{
cout << "The strings are NOT equal" << endl;
}

return 0;
}
```

Output

```
Compare Strings using Strcmp() function

Insert first string:
Ejike
Insert second string:
Ejike
The strings are EQUAL

Process returned 0 (0x0)   execution time : 6.169 s
Press any key to continue.
```

Practice 69: A Program to Copy Strings

```
#include <iostream>

using namespace std;

int main()
{
char str1[100], str2[100], x;
cout << "Copy strings using for loop \n" << endl;

cout << "Insert a string: " << endl;
cin >> str1;

for(x=0; str1[x]!='\0'; ++x)
{
str2[x] = str1[x];
}

str2[x] = '\0';

cout << "String copied successfully... " << str2 << endl;

return 0;
}
```

Output

```
Copy strings using for loop

Insert a string:
Ejike
String copied successfully... Ejike

Process returned 0 (0x0)   execution time : 5.166 s
Press any key to continue.
```

Practice 70: A Program to Copy Strings using strcpy

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str1[100],str2[100];
cout << "Copy Strings using Strcpy function \n" << endl;

cout << "Insert some strings: " << endl;
cin >> str1;

strcpy(str2,str1);

cout << "String coped successfully..." << str2 << endl;

return 0;
}
```

Output

```
Copy Strings using Strcpy function

Insert some strings:
IfeanyiChukwu
String coped successfully...IfeanyiChukwu

Process returned 0 (0x0)   execution time : 13.437 s
Press any key to continue.
```

Practice 71: A Program for Concatenation using a for loop

```
#include <iostream>

using namespace std;

int main()
{
char str1[100], str2[100];
int x,y;
cout << "Concatenation using a for loop \n" << endl;

cout << "Insert first string: " << endl;
cin >> str1;

cout << "Insert second string: " << endl;
cin >> str2;

for(x=0; str1[x]!='\0'; ++x);

for(y=0; str2[y]!='\0'; ++y,++x)
{
str1[x]=str2[y];
}

str1[x] = '\0';

cout << "String concatenated successfully..." << str1 << endl;

return 0;
}
```

Output

```
Concatenation using a for loop

Insert first string:
Ejike
Insert second string:
IfeanyiChukwu
String concatenated successfully...EjikeIfeanyiChukwu

Process returned 0 (0x0)   execution time : 12.937 s
Press any key to continue.
```

Practice 72: A Program to Concatenation using strcat

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str1[100], str2[100];
cout << "String concatenation using Strcat Function \n" << endl;

cout << "Insert first string: " << endl;
cin >> str1;

cout << "Insert second string: " << endl;
cin >> str2;

strcat(str1,str2);

cout << "String concatenated successfully..." << str1 << endl;

return 0;
}
```

Output

```
String concatenation using Strcat Function

Insert first string:
Ejike
Insert second string:
IfeanyiChukwu
String concatenated successfully...EjikeIfeanyiChukwu

Process returned 0 (0x0)   execution time : 9.873 s
Press any key to continue.
```

Practice 73: A Program to Reverse a String using while loop

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[100], rev;
int x,y=0;
cout << "Reverse a string using while loop \n" << endl;

cout << "Insert a string: " << endl;
cin >> str;

x=0;
y= strlen(str) - 1;

while(x<y)
{
rev = str[x];
str[x] = str[y];
str[y] = rev;
x++;
y--;
}

cout << "The reversed string = " << str << endl;

return 0;
}
```

Output

```
Reverse a string using while loop

Insert a string:
Ejike
The reversed string  = ekijE

Process returned 0 (0x0)   execution time : 3.657 s
Press any key to continue.
```

Practice 74: A Program to Reverse a String using strrev

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[100];
cout << "Reverse a string using Strrev Function \n" << endl;

cout << "Insert a string: " << endl;
cin >> str;

strrev(str);

cout << "The reversed string is : " << str << endl;

return 0;
}
```

Output

```
Reverse a string using Strrev Function

Insert a string:
IfeanyiChukwu
The reversed string is : uwkuhCiynaefI

Process returned 0 (0x0)   execution time : 9.646 s
Press any key to continue.
```

Practice 75: A Program to Reverse Complete Sentence

```
#include <iostream>
#include <string.h>
using namespace std;

void rev_sentence(const string&str);

int main()
{
string str;
cout << "How to reverse a sentence \n" << endl;

cout << "Insert some sentences: " << endl;
getline(cin,str);

rev_sentence(str);
return 0;
}

void rev_sentence(const string&str)
{
size_t n = str.size();

if(n == 1)
{
cout << str << endl;
}
else
{
cout << str[n-1];
string rev_str = str.substr(0,n-1);
rev_sentence(rev_str);
}
}
```

Output

```
How to reverse a sentence

Insert some sentences:
Welcome to C Plus Plus Programming Masterclass
ssalcretsaM gnimmargorP sulP sulP C ot emocleW

Process returned 0 (0x0)   execution time : 39.852 s
Press any key to continue.
```

Practice 76: A Program to Check if a String is a Palindrome

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[50];
int x, len, flag=0;
cout << "Check if a string is a Palindrome \n" << endl;

cout << "Insert a string: " << endl;
cin >> str;

len = strlen(str);

for(x=0; x<len; x++)
{
if(str[x]!=str[len-x-1])
{
flag = 1;
break;
}
}

if(flag)
{
cout << str << " is NOT a palindrome!" << endl;
}
else
{
cout << str << " is a PALINDROME!" << endl;

}

return 0;
}
```

Output

```
Check if a string is a Palindrome

Insert a string:
heolloeh
heolloeh is a PALINDROME!

Process returned 0 (0x0)   execution time : 14.102 s
Press any key to continue.
```

Practice 77: A Program to Convert from Uppercase to Lowercase

```
#include <iostream>

using namespace std;

void case_lower(char str[]);

int main()
{
char str[200];
cout << "Uppercase to Lowercase using a while loop \n" << endl;

cout << "Insert uppercase strings to convert to lowercase: " <<endl;
cin >> str;

case_lower(str);

cout << "The result of the lowercase conversion is: " << str << endl;

return 0;
}

void case_lower(char str[])
{
int x = 0;
while(str[x]!='\0')
{
if(str[x]>='A' && str[x]<='Z')
{
str[x] = str[x]+32;
}
x++;
}
}
```

Output

```
Uppercase to Lowercase using a while loop

Insert uppercase strings to convert to lowercase:
PROGRAMMING
The result of the lowercase conversion is: programming

Process returned 0 (0x0)   execution time : 4.881 s
Press any key to continue.
```

Practice 78: A Program to Convert from Uppercase to Lowercase using strlwr

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[100];
cout << "Uppercase to Lowercase using strlwr function \n" << endl;

cout << "Insert Uppercase string to convert to lowercase: " << endl;
cin >> str;

strlwr(str);

cout << "The result of the string conversion is: " << str << endl;

return 0;
}
```

Output

```
Uppercase to Lowercase using strlwr function

Insert Uppercase string to convert to lowercase:
HELLOWORLD
The result of the string conversion is: helloworld

Process returned 0 (0x0)   execution time : 24.327 s
Press any key to continue.
```

Practice 79: A Program to Convert Uppercase to Lowercase using while loop

```
#include <iostream>
using namespace std;

void case_upper(char str[]);

int main()
{
char str[200];
cout << "Lowercase to Uppercase using while loop \n" << endl;

cout << "Insert lowercase string to convert to uppercase: " << endl;
cin >> str;

case_upper(str);

cout << "The conversion of lowercase to uppercase is: " << str << endl;

return 0;
}

void case_upper(char str[])
{
int x=0;

while(str[x]!='\0')
{
if(str[x]>='a' && str[x]<='z')
{
str[x] = str[x]-32;
}
x++;
}
}
```

Output

```
Lowercase to Uppercase using while loop

Insert lowercase string to convert to uppercase:
helloworld
The conversion of lowercase to uppercase is: HELLOWORLD

Process returned 0 (0x0)   execution time : 8.873 s
Press any key to continue.
```

Practice 80: A Program to Convert Lowercase to Uppercase using strupr

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[200];
cout << "Lowercase to Uppercase using strupr function \n" <<
endl;

cout << "Insert Uppercase string for conversion: " << endl;
cin >> str;

strupr(str);

cout << "The result of conversion to uppercase is: " << str <<
endl;

return 0;
}
```

Output

```
Lowercase to Uppercase using strupr function

Insert Uppercase string for conversion:
ifeanyichukwu
The result of conversion to uppercase is: IFEANYICHUKWU

Process returned 0 (0x0)   execution time : 14.601 s
Press any key to continue.
```

Practice 81: A Program to Remove Vowels from String

```
#include <iostream>
#include <string.h>

using namespace std;

int main()
{
char str[200];
int len, x, y;
cout << "Delete vowels from a string \n" << endl;

cout << "Insert a string: " << endl;
cin >> str;

len = strlen(str);

for(x=0; x<len; x++)
{
if(str[x]=='a' || str[x]=='A' || str[x]=='e' ||str[x]=='E'
|| str[x]=='o'|| str[x]=='O' || str[x]=='u' || str[x]=='U'
|| str[x]=='i' || str[x]=='I')
{
for(y=x; y<len; y++)
{
str[y] = str[y+1];
}
len--;
}
}

cout << "String after vowel removal is: " << str << endl;

return 0;
}
```

Output

```
Delete vowels from a string

Insert a string:
Helloworld
String after vowel removal is: Hllwrld

Process returned 0 (0x0)   execution time : 14.129 s
Press any key to continue.
```

Practice 82: A Program to Find the Frequency Occurrence of a Character

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
char str[200], ch;
int x, cnt=0;
cout << "Character Frequency in a sentence \n" << endl;

cout << "Insert some string: " << endl;
cin.getline(str,1000);

cout << "Choose a character to find its number of occurances: " << endl;
cin >> ch;

for(x=0; str[x]!='\0'; ++x)
{
if(ch == str[x])
++cnt;
}

cout << "The frequency of character occurance is: " << cnt << endl;

return 0;
}
```

Output

```
Character Frequency in a sentence

Insert some string:
Ifeanyichukwu
Choose a character to find its number of occurances:
u
The frequency of character occurance is: 2

Process returned 0 (0x0)   execution time : 32.041 s
Press any key to continue.
```

Practice 83: A Program to Check Vowels and Consonants

```
#include <iostream>

using namespace std;

int main()
{
char ch;
cout << "Check for Vowel and Consonant \n" << endl;

cout << "Insert a character: " << endl;
cin >> ch;

if(ch=='a'||ch=='A'||ch=='e'||ch=='E'||ch=='i'||ch=='I'
||ch=='o'||ch=='O'||ch=='u'||ch=='U')
{
cout << "The character " << ch << " is a VOWEL!" << endl;
}
else
{
cout << "The character " << ch << " is a CONSONANT!" << endl;
}

return 0;
}
```

Output

```
Check for Vowel and Consonant

Insert a character:
P
The character P is a CONSONANT!

Process returned 0 (0x0)   execution time : 5.839 s
Press any key to continue.
```

Practice 84: A Program to Get Number of Vowels, Consonants, Digits & Whitespaces in a Sentence

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
char str[200];
int x, vol=0, con=0, dig=0, spa=0;
cout << "Number of Vowels, Consonants, Digits & WhiteSpaces \n" << endl;
cout << " Insert Some strings: " << endl;
cin.getline(str,1000);

for(x=0; str[x]!='\0'; ++x)
{
if(str[x]=='a'||str[x]=='A'||str[x]=='e'||str[x]=='E'||str[x]=='i'||str[x]=='
I'
||str[x]=='o'||str[x]=='O'||str[x]=='u'||str[x]=='U')
{
++vol;
}
else if(str[x]>='a' && str[x]<='z' || str[x]>='A' && str[x]<='Z')
{
++con;
}
else if(str[x]>='0' && con <='9')
{
++dig;
}
else if(str[x]==' ')
{
++spa;
}
}
cout << "\n Number of Vowels is: " << vol << endl;
cout << "\n Number of Consonants is: " << con << endl;
cout << "\n Number of Digits is: " << dig << endl;
cout << "\n Number of White Spaces is: " << spa << endl;
return 0;
}
```

Output

```
Number of Vowels, Consonants, Digits & WhiteSpaces

 Insert Some strings:
Hello World 123 Ejike

 Number of Vowels is: 6

 Number of Consonants is: 9

 Number of Digits is: 3

 Number of White Spaces is: 3

Process returned 0 (0x0)   execution time : 12.973 s
Press any key to continue.
```

Practice 85: A Program to Search for an Element in an Array

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, n, s, amt[200];
cout << "Search for Element in an Array \n" << endl;

cout << "Insert range of array elements: " << endl;
cin >> n;

cout << "Insert elements of array: " <<endl;
for(int a=0; a<n; a++)
{
cin >> amt[a];
}

cout << "Choose an element to find within the array: " << endl;
cin >> s;

x = 0;
y = n-1;
z = (x+y)/2;

while(x <= y)
{
if(amt[z] == s)
{
cout << s << " found at location " << z+1 << endl;
break;
}
else if(amt[z] < s)
{
x = z+1;
}
else
{
y = z + 1;
z = (x+y)/2;
}
}

if(x > y)
{
cout << s << " is not found in the array list!" << endl;
}

return 0;
}
```

Output

```
Search for Element in an Array

Insert range of array elements:
4
Insert elements of array:
45
67
89
34
Choose an element to find within the array:
67
67 found at location 2

Process returned 0 (0x0)   execution time : 24.650 s
Press any key to continue.
```

Practice 86: A Program to Check for an Element in an Array using Recursive Function

```
#include <iostream>

using namespace std;

int arr_search(int amt[], int x, int y, int a, int b);

int main()
{
int amt[50], x, y, z, output, a, b;
cout << "Search element of an Array using Recursive \n" << endl;

cout << "Insert range of array elements: " << endl;
cin >> x;

cout << "Insert elements of the array: " << endl;
for(y=0; y<x; y++)
{
cin >> amt[y];
}

cout << "Insert the element to search within the array: " << endl;
cin >> z;

a=0, b = x-1;

output = arr_search(amt, x, z, a, b);

if(output == 0)
{
cout << "The Number is found " << endl;
}
else
{
cout << "The Number is not found. " << endl;
}

return 0;
}

int arr_search(int amt[], int x, int y, int a, int b)
{
int z, output=0;
if(a <= b)
{
z = (a+b)/2;
if(y ==amt[z])
{
output = 1;
}
else if(y<amt[z])
{
return arr_search(amt,x,y,a,z-1);
}
```

```
else
{
return arr_search(amt,x,y,z+1,b);
}
}
else
{
return output;
}
}
```

Output

```
Search element of an Array using Recursive

Insert range of array elements:
4
Insert elements of the array:
325678
90
12
32
Insert the element to search within the array:
12
The Number is found

Process returned 0 (0x0)   execution time : 19.658 s
Press any key to continue.
```

Practice 87: A Program to Sort Array using Bubble

```
#include <iostream>

using namespace std;

void sort_bubble(int amt[], int n);

int main()
{
int amt[200], n, x, sw;
cout << "Sorting of Array elements using Bubble method \n" << endl;

cout << "Insert range of array elements: " << endl;
cin >> n;

cout << "Insert elements of the array: " << endl;
for(x=0; x<n; x++)
{
cin >> amt[x];
}

sort_bubble(amt,n);

cout << "The sorted array is: " << endl;
for(x=0; x<n; x++)
{
cout << amt[x] << endl;
}

return 0;
}

void sort_bubble(int amt[], int n)
{
int x;
for(int y=0; y<(n-1); y++)
{
for(int z=0; z<n-y-1; z++)
{
if(amt[z] > amt[z+1])
{
x = amt[z];
amt[z] = amt[z+1];
amt[z+1] = x;
}
}
}
}
```

Output

```
Sorting of Array elements using Bubble method

Insert range of array elements:
4
Insert elements of the array:
67
87
98
32
The sorted array is:
32
67
87
98

Process returned 0 (0x0)   execution time : 15.211 s
Press any key to continue.
```

Practice 88: A Program to Sort Arrays using Insertion

```
#include <iostream>
using namespace std;

int main()
{
int n, amt[200], x, y, z;
cout << "Sorting Array elements using Insertion \n" << endl;
cout << "Choose range of array elements: " << endl;
cin >> n;
cout << "Insert elements of the array: " << endl;
for(x=0; x<n; x++)
{
cin >> amt[x];
}

for(x=1; x<=n-1; x++)
{
y=x;

while(y>0 && amt[y] < amt[y-1])
{
z = amt[y];
amt[y] = amt[y-1];
amt[y-1] = z;
y--;
}
}

cout << "The sorted array is: " << endl;
for(x=0; x<=n-1; x++)
{
cout << amt[x] << endl;
}
return 0;
}
```

Output

```
Sorting Array elements using Insertion

Choose range of array elements:
4
Insert elements of the array:
43
21
54
78
The sorted array is:
21
43
54
78
```

Practice 89: A Program to Sort Arrays using Selection Method

```
#include <iostream>
using namespace std;

int main()
{
int amt[200], n, x, y, pos, z;
cout << "Sorting array elements using selection \n" << endl;
cout << "Insert range of array element: " << endl;
cin >> n;
cout << "Insert the elements of the array: " << endl;
for(x=0; x<n; x++)
{
cin >> amt[x];
}

for(x=0; x<(n-1); x++)
{
pos = x;
for(y=x+1; y<n; y++)
{
if(amt[pos] > amt[y])
{
pos = y;
}
}
if(pos != x)
{
z = amt[x];
amt[x] = amt[pos];
amt[pos] = z;
}
}

cout << "The result of the sorted array is: " << endl;
for(x=0; x<n; x++)
{
cout << amt[x] << endl;
}
return 0;
}
```

Output

```
Sorting array elements using selection

Insert range of array element:
3
Insert the elements of the array:
47
25
8
The result of the sorted array is:
8
25
47

Process returned 0 (0x0)   execution time : 17.032 s
Press any key to continue.
```

Practice 90: A Program to Call by Value

```
#include <iostream>

using namespace std;

int call_value(int a, int b);

int main()
{
int x, y, sum;
cout << "Call by Value \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << " Insert second number: " << endl;
cin >> y;

sum = call_value(x,y);

cout << "The sum of " << x << " and " << y << " is: " << sum << endl;

return 0;
}

int call_value(int a, int b)
{
int sum;
sum = a+b;

return sum;
}
```

Output

```
Call by Value

Insert first number:
79
 Insert second number:
65
The sum of 79 and 65 is: 144

Process returned 0 (0x0)   execution time : 8.437 s
Press any key to continue.
```

Practice 91: A Program to Call by Reference

```
#include <iostream>

using namespace std;

void ref_swap(int &x, int &y);

int main()
{
int x, y;
cout << "Call by Reference \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

cout << "Numbers before swapping are: " << x << " and " << y << endl;

ref_swap(x,y);

cout << "Numbers after swapping are: " << x << " and " << y << endl;

return 0;
}

void ref_swap(int &x, int &y)
{
int z;

z = x;
x = y;
y = z;

return;
}
```

Output

```
Call by Reference

Insert first number:
70
Insert second number:
12
Numbers before swapping are: 70 and 12
Numbers after swapping are: 12 and 70

Process returned 0 (0x0)   execution time : 6.551 s
Press any key to continue.
```

Practice 92: A Program to Call by Pointer

```
#include <iostream>

using namespace std;

void point_swap(int *x, int *y);

int main()
{
int x, y;
cout << "Call by Pointer \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

cout << "Numbers before swapping are " << x << " and " << y << endl;

point_swap(&x, &y);

cout << "Numbers after swapping are " << x << " and " << y << endl;

return 0;
}

void point_swap(int *x, int *y)
{
int z;

z = *x;
*x = *y;
*y = z;
}
```

Output

```
Call by Reference

Insert first number:
67
Insert second number:
34
Numbers before swapping are: 67 and 34
Numbers after swapping are: 34 and 67

Process returned 0 (0x0)   execution time : 4.119 s
Press any key to continue.
```

Practice 93: A Program to Write to a File

```
#include <iostream>
#include <fstream>
#include <cstdlib>

using namespace std;

int main()
{
char data[500], ch;
cout << "Static Storage Class \n" << endl;

ofstream fout("C:/Users/USER/Desktop/CFiles/C++.txt",ios::out);

cout << "Insert some text into the file: " << endl;
for(int x=0; x<4; x++)
{
cin.get(data,500);
cin.get(ch);
fout << data << endl;
}

fout.close();

return 0;
}
```

Output

```
Static Storage Class

Insert some text into the file:
Hello World
```

Practice 94: A Program to Read from a File

```
#include <iostream>
#include <fstream>
#include <cstdlib>

using namespace std;

int main()
{
char data[500], ch;

cout << "File Handling..." << endl;

ifstream fin("C:/Users/USER/Desktop/CFiles/C++.txt",ios::in);
fin.seekg(0);

cout << "\n" << endl;
for(int x=0; x<5; x++)
{
fin.get(data,500);
fin.get(ch);
cout << data << "\n" << endl;
}

fin.close();

return 0;
}
```

Output

```
File Handling...


Hello World




Process returned 0 (0x0)   execution time : 0.376 s
Press any key to continue.
```

Practice 95: A Program to Add 2 Numbers using Function

```
#include <iostream>

using namespace std;

int addition(int x, int y);

int main()
{
int x, y, sum;
cout << "Add two numbers using function \n" << endl;

cout << "Insert first numbers: " << endl;
cin >> x;

cout << "Insert second numbers: " << endl;
cin >> y;

sum = addition(x,y);

cout << "The addition of " << x << " and " << y << " is " << sum << endl;

return 0;
}

int addition(int x, int y)
{
int sum;
sum = x+y;
return sum;
}
```

Output

```
Add two numbers using function

Insert first numbers:
29
Insert second numbers:
34
The addition of 29 and 34 is 63

Process returned 0 (0x0)   execution time : 8.650 s
Press any key to continue.
```

Practice 96: A Program to Subtract Two Numbers using Function

```
#include <iostream>

using namespace std;

int subtraction(int x, int y);

int main()
{
int x, y, sub;
cout << "Subtraction of two numbers using a function \n" << endl;

cout << "Insert first number: " << endl;
cin >> x;

cout << "Insert second number: " << endl;
cin >> y;

sub = subtraction(x,y);

cout << "The result of subtraction of " << y << " from " << x << " is " <<
sub << endl;

return 0;
}

int subtraction(int x, int y)
{
int sub;
sub = x - y;
return sub;
}
```

Output

```
Subtraction of two numbers using a function

Insert first number:
12
Insert second number:
89
The result of subtraction of 89 from 12 is -77

Process returned 0 (0x0)   execution time : 7.785 s
Press any key to continue.
```

Practice 97: A Program to Create Classes

```
#include <iostream>

using namespace std;

class Area
{
public:
int length;
int breadth;
};

int main()
{
Area A;
int area;
cout << "Simple class creations \n" << endl;

cout << "Insert the length of the rectangle: " << endl;
cin >> A.length;

cout << "Insert the breadth of the rectangle: " << endl;
cin >> A.breadth;

area = A.length*A.breadth;
cout << "The area of the rectangle is: " << area << endl;

return 0;
}
```

Output

```
Simple class creations

Insert the length of the rectangle:
17
Insert the breadth of the rectangle:
6
The area of the rectangle is: 102

Process returned 0 (0x0)   execution time : 11.919 s
Press any key to continue.
```

Practice 98: A Program to Pass Argument to Function

```
#include <iostream>
using namespace std;

void show(char='A', int=20);

int main()
{
cout << "Passing Default Argument to a function \n" << endl;

cout << "Argument is not passed here:" << endl;
show();
cout << "Argument is passed here:" << endl;
show('B');
cout << "Argument is not passed here:" << endl;
show('B',10);
return 0;
}

void show(char ch, int x)
{
for(int y=1; y<=x; ++y)
{
cout << ch << endl;
}
cout << endl;
}
```

Output

```
Passing Default Argument to a function

Argument is not passed here:
A
A
A
A
A

Argument is passed here:
B
B
B
B
B

Argument is not passed here:
B
B
B


Process returned 0 (0x0)   execution time : 0.183 s
Press any key to continue.
```

Practice 99: A Program to use Inline Function

```
#include <iostream>

using namespace std;

inline int additon(int a, int b)
{
int c = a + b;
return (c);
}

int main()
{
cout << "Inline Function \n" << endl;

cout << "The addition of numbers is " << additon(13,8) << endl;

cout << "The addition of numbers is " << additon(67,5) << endl;

return 0;
}
```

Output

```
Inline Function

The addition of numbers is 21
The addition of numbers is 72

Process returned 0 (0x0)   execution time : 0.421 s
Press any key to continue.
```

Practice 100: A Program to Check Length of String using Pointer

```
#include <iostream>

using namespace std;

int len_str(char*);

int main()
{
char str[100];
int len;
cout << "Length of string using pointer \n" << endl;

cout << "Insert any string to calculate its length: " << endl;
cin >> str;

len = len_str(str);

cout << "The length of the string is: " << len << endl;

return 0;
}

int len_str(char*ptr)
{
int cnt = 0;
while(*ptr != '\0')
{
cnt++;
ptr++;
}
return cnt;
}
```

Output

```
Length of string using pointer

Insert any string to calculate its length:
HelloWorld
The length of the string is: 10

Process returned 0 (0x0)   execution time : 5.306 s
Press any key to continue.
```

Practice 101: A Program to Create a Pyramid Pattern

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, n;
cout << "Pyramid Pattern \n" << endl;

cout << "Insert number of rows: " << endl;
cin >> n;

z = n;

for(x=1; x<=n; x++)
{
for(y=1; y<z; y++)
{
cout << " ";
}
z--;
for(y=1; y<=2*x-1; y++)
{
cout << "*";
}
cout << endl;
}

return 0;
}
```

Output

```
Pyramid Pattern

Insert number of rows:
6
     *
    ***
   *****
  *******
 *********
***********

Process returned 0 (0x0)   execution time : 5.066 s
Press any key to continue.
```

Practice 102: A Program to Create a Right-Angle Triangle

```
#include <iostream>

using namespace std;

int main()
{
int x, y, n;
cout << "Right Angle Triangle \n" << endl;

cout << "Insert number of rows of triangle: " << endl;
cin >> n;

for(x=1; x<=n; ++x)
{
for(y=1; y<=x; ++y)
{
cout << "*";
}
cout << endl;
}

return 0;
}
```

Output

```
Right Angle Triangle

Insert number of rows of triangle:
8
*
**
***
****
*****
******
*******
********

Process returned 0 (0x0)   execution time : 4.219 s
Press any key to continue.
```

Practice 103: A Program to a Diamond Pattern

```
#include <iostream>
using namespace std;

int main()
{
int n,x,y,z=1;
cout << "Diamond Pattern \n" << endl;
cout << "Insert number of rows: " << endl;
cin >> n;

z = n-1;

for(x=1; x<=n; x++)
{
for(y=1; y<=z; y++)
{
cout << " ";
}
z--;
for(y=1; y<=2*x-1;y++)
{
cout << "*";
}
cout << endl;
}
z = 1;

for(x=1; x<=n-1; x++)
{
for(y=1; y<=z; y++)
{
cout << " ";
}
z++;
for(y=1; y<=2*(n-x)-1; y++)
{
cout << "*";
}
cout << endl;
}
return 0;
}
```

Output

```
Diamond Pattern

Insert number of rows:
4
   *
  ***
 *****
*******
 *****
  ***
   *

Process returned 0 (0x0)   execution time : 3.126 s
Press any key to continue.
```

Practice 104: A Program to Create a Right-Angle Triangle Pattern

```
#include <iostream>

using namespace std;

int main()
{
int x, y, n;
cout << "Numeric Right Angle Triangle \n" << endl;

cout << "Insert number of rows: " << endl;
cin >> n;

for(x=1; x<=n; ++x)
{
for(y=1; y<=x; ++y)
{
cout << y << " ";
}
cout << endl;
}

return 0;
}
```

Output

```
Numeric Right Angle Triangle

Insert number of rows:
10
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10

Process returned 0 (0x0)   execution time : 1.812 s
Press any key to continue.
```

Practice 105: A Program to Create Alphabet Pattern

```
#include <iostream>

using namespace std;

int main()
{
char ch, data='A';
cout << "Alphabet Right Angle Triangle \n" << endl;

cout << "Insert an uppercase character of the last alphabet: " << endl;
cin >> ch;

for(int x=1; x<=(ch-'A'+1); ++x)
{
for(int y=1; y<=x; ++y)
{
cout << data << " ";
}
++data;

cout << endl;
}

return 0;
}
```

Output

```
Alphabet Right Angle Triangle

Insert an uppercase character of the last alphabet:
Z
A
B B
C C C
D D D D
E E E E E
F F F F F
G G G G G G
H H H H H H H
I I I I I I I I
J J J J J J J J J
K K K K K K K K K K
L L L L L L L L L L L
M M M M M M M M M M M M
N N N N N N N N N N N N N
O O O O O O O O O O O O O O
P P P P P P P P P P P P P P P
Q Q Q Q Q Q Q Q Q Q Q Q Q Q Q
R R R R R R R R R R R R R R R R
S S S S S S S S S S S S S S S S S
T T T T T T T T T T T T T T T T T T
U U U U U U U U U U U U U U U U U U U
V V V V V V V V V V V V V V V V V V V V
W W W W W W W W W W W W W W W W W W W W W
X X X X X X X X X X X X X X X X X X X X X X
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z Z

Process returned 0 (0x0)   execution time : 3.212 s
Press any key to continue.
```

Practice 106: A Program to Create a Pascal Triangle Pattern

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z, n, num=1;
cout << "Paschal Triangle \n" << endl;

cout << "Insert number of rows: " << endl;
cin >> n;

for(x=0; x<n; x++)
{
for(y=1; y<=n-x; y++)
{
cout << " ";
}

for(z=0; z<=x; z++)
{
if(z==0 || x==0)
{
num=1;
}
else
{
num = num*(x-z+1)/z;
}
cout << " " << num;
}
cout << endl;
}

return 0;
}
```

Output

```
Paschal Triangle

Insert number of rows:
6
       1
      1 1
     1 2 1
    1 3 3 1
   1 4 6 4 1
  1 5 10 10 5 1

Process returned 0 (0x0)   execution time : 2.812 s
Press any key to continue.
```

Practice 107: A Program to Create a Floyd's Triangle

```
#include <iostream>

using namespace std;

int main()
{
int x, y, z=1, n;
cout << "Floyd's Triangle \n" << endl;

cout << "Insert number of rows: " << endl;
cin >> n;

for(x=1; x<=n; x++)
{
for(y=1; y<=x; y++)
{
cout << z << " ";
z++;
}
cout << endl;
}

return 0;
}
```

Output

```
Floyd's Triangle

Insert number of rows:
7
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
22 23 24 25 26 27 28

Process returned 0 (0x0)   execution time : 6.282 s
Press any key to continue.
```

Practice 108: A Program to Create Numeric Pyramid Pattern

```
#include <iostream>
using namespace std;

int main()
{
int x, y, n, k=0, cnt=0, cnt1=0;
cout << "Numeric Pyramid Pattern \n" << endl;
cout << "Insert number of rows: " << endl;
cin >> n;

for(x=1; x<=n; ++x)
{
for(y=1; y<=n-x; ++y)
{
cout << " ";
++cnt;
}

while(k!=2*x-1)
{
if(cnt<=n-1)
{
cout << x+k << " ";
++cnt;
}
else
{
++cnt1;
cout << x+k-2*cnt1 << " ";
}
++k;
}
cnt1=cnt=k=0;
cout << endl;
}
return 0;
}
```

Output

```
Numeric Pyramid Pattern

Insert number of rows:
4
   1
  2 3 2
 3 4 5 4 3
4 5 6 7 6 5 4

Process returned 0 (0x0)   execution time : 3.453 s
Press any key to continue.
```