# **Problem #6 to #10 Solutions**

—------------------------------------------------------------------------------------

# **Problem 6**

====================

Description: Write A program to get the full name:

=========================================

#include <iostream>

using namespace std;

struct stInfo{

string first\_name;

string last\_name;

};

stInfo ReadInfo(){

stInfo info;

cout<<"\nEnter The First Name: ";

cin>>info.first\_name;

cout<<"\nEnter The Last Name: ";

cin>> info.last\_name;

return info;

}

string GetFullName(stInfo info){

string full\_name ="";

full\_name = info.first\_name +" "+ info.last\_name;

return full\_name;

}

void PrintFullName(string full\_name){

cout<<"\nYour Full name is: "<<full\_name<<endl;

}

int main(){

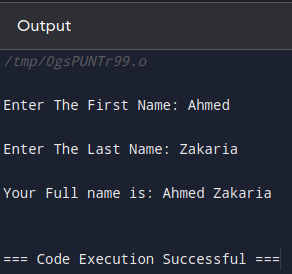
PrintFullName(GetFullName(ReadInfo()));

}

===========================

The output:

============



========================================================================

# **Problem 7**

==========================

Description: Write A program to print a half a number;

==========================================

#include <iostream>

using namespace std;

int ReadNumber(){

int Number = 0;

cout<<"\nEnter Number: ";

cin>> Number;

return Number;

}

float CalculateHalfNumber(int Num){

return (float) Num /2;

}

void PrinResults(int Num){

string result="Half of "+ to\_string(Num)+" is "+ to\_string(CalculateHalfNumber(Num));

cout<<endl<<result<<endl;

}

int main(){

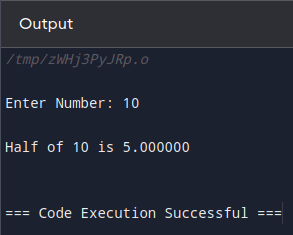
PrinResults(ReadNumber());

}

==============================================================

The Output:

===========



========================================================================

# **Problem 8**

=======================

Description: Write a Program to: Mark Pass Fail

=======================================

#include <iostream>

using namespace std;

enum enPassFail{ Pass = 1, Fail = 2};

int ReadMark(){

int Mark = 0;

cout<<"\nEnter Your Mark: ";

cin>> Mark;

return Mark;

}

enPassFail CheckMark(int Mark){

if(Mark >= 50)

return enPassFail::Pass;

else

return enPassFail::Fail;

}

void PrintResults(int Mark){

if(CheckMark(Mark)==enPassFail::Pass)

cout<<"\nYou Passed";

else

cout<<"\nYou Failed";

}

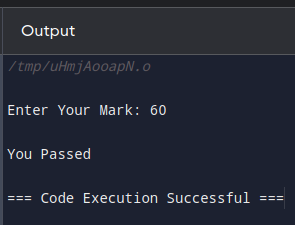
int main(){

PrintResults(ReadMark());

}

=====================================================

The Output:



=======================================================================

# **Problem 9**

================================

Description: Write a Program to get Sum of Three Numbers;

==================================================

#include <iostream>

using namespace std;

void ReadNumbers(int& N1, int& N2,int& N3){

cout<<"\nEnter Number 1: ";

cin>> N1;

cout<<"\nEnter Number 2: ";

cin>> N2;

cout<<"\nEnter Number 3: ";

cin>> N3;

}

int SumOf3Numbers(int N1 , int N2, int N3){

return N1 + N2 + N3;

}

void PrintResults(int Total){

cout<<"\nThe Total sum of The three Numbers is: "<<Total<<endl;

}

int main(){

int N1, N2 , N3;

ReadNumbers(N1,N2,N3);

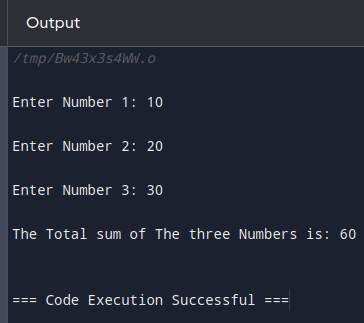
PrintResults(SumOf3Numbers(N1,N2,N3));

}

========================================================================

The Output:

============



========================================================================

# **Problem 10**

======================

Description: Write A Program To Get Average Of Three Marks;

===================================================

#include <iostream>

using namespace std;

void ReadNumbers(int& Mark1, int& Mark2,int& Mark3){

cout<<"\nEnter Number 1: ";

cin>> Mark1;

cout<<"\nEnter Number 2: ";

cin>> Mark2;

cout<<"\nEnter Number 3: ";

cin>> Mark3;

}

int SumOf3Numbers(int Mark1 , int Mark2, int Mark3){

return Mark1 + Mark2 + Mark3;

}

float CalculateAverage(int Mark1 , int Mark2, int Mark3){

return (float)SumOf3Numbers(Mark1, Mark2, Mark3) / 3;

}

void PrintResults(float Average){

cout<<"\nThe Average Of the 3 Numbers is: "<<Average<<endl;

}

int main(){

int Mark1, Mark2 , Mark3;

ReadNumbers(Mark1,Mark2,Mark3);

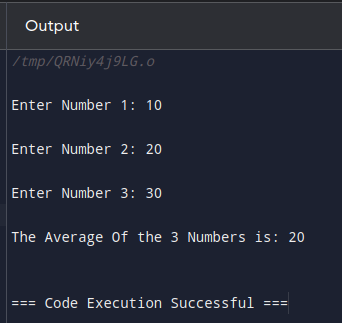
PrintResults(CalculateAverage(Mark1,Mark2,Mark3));

}

========================================================================

The Output:

===========



========================================================================