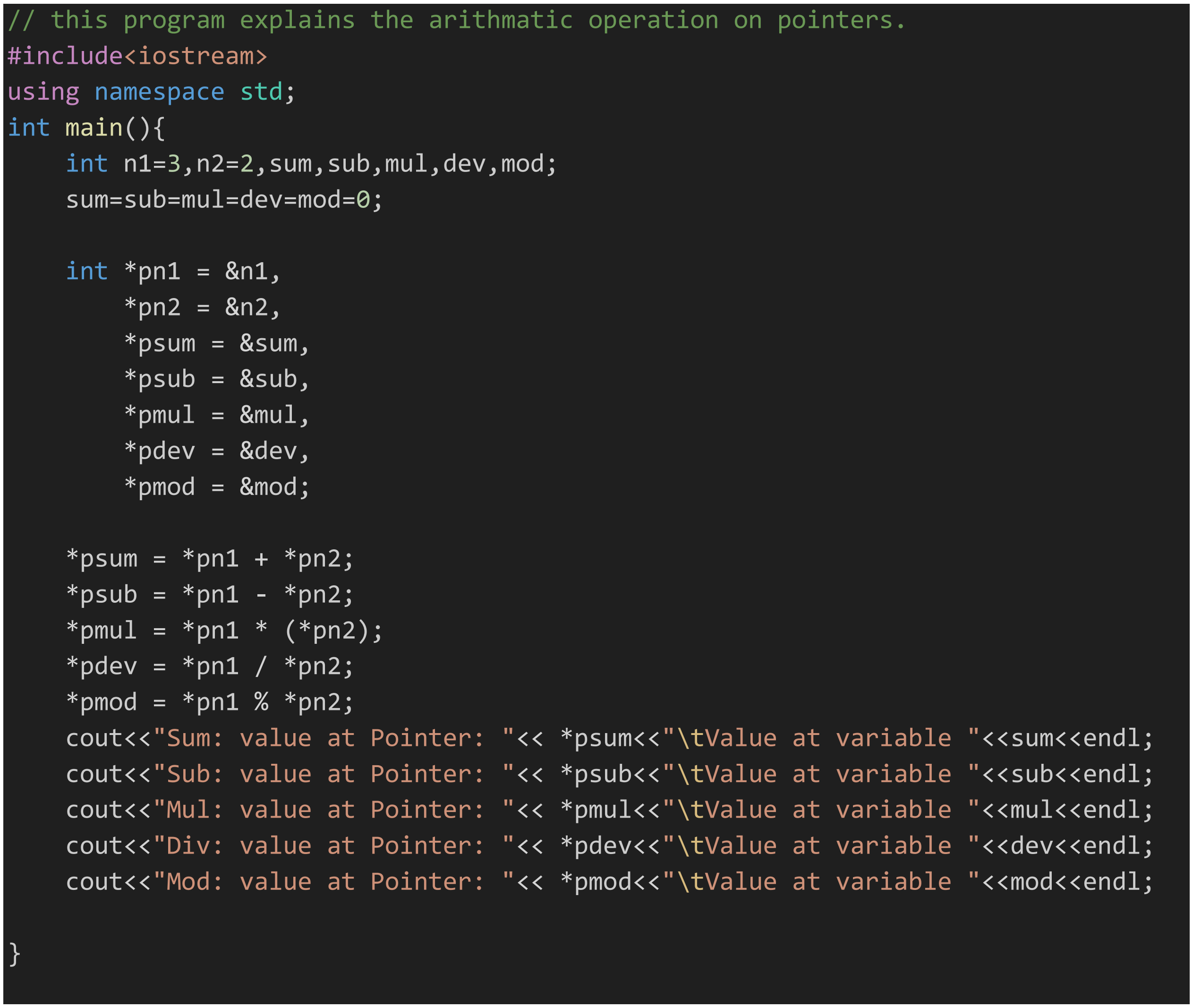
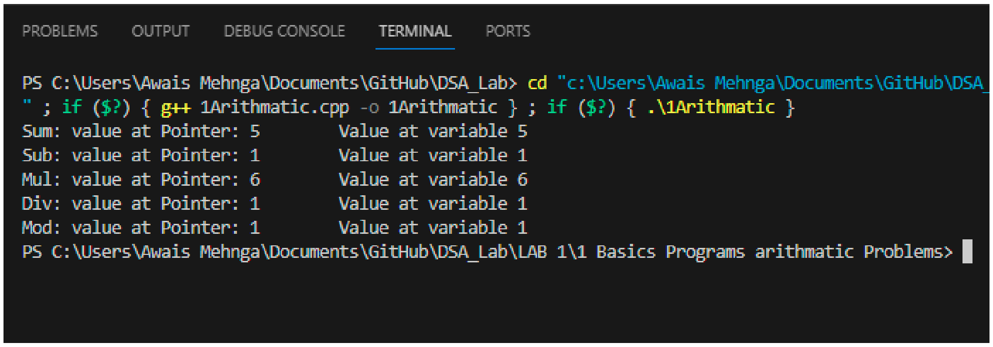
DSA LAB 1 Assignment Programs

Pointers

Program 1

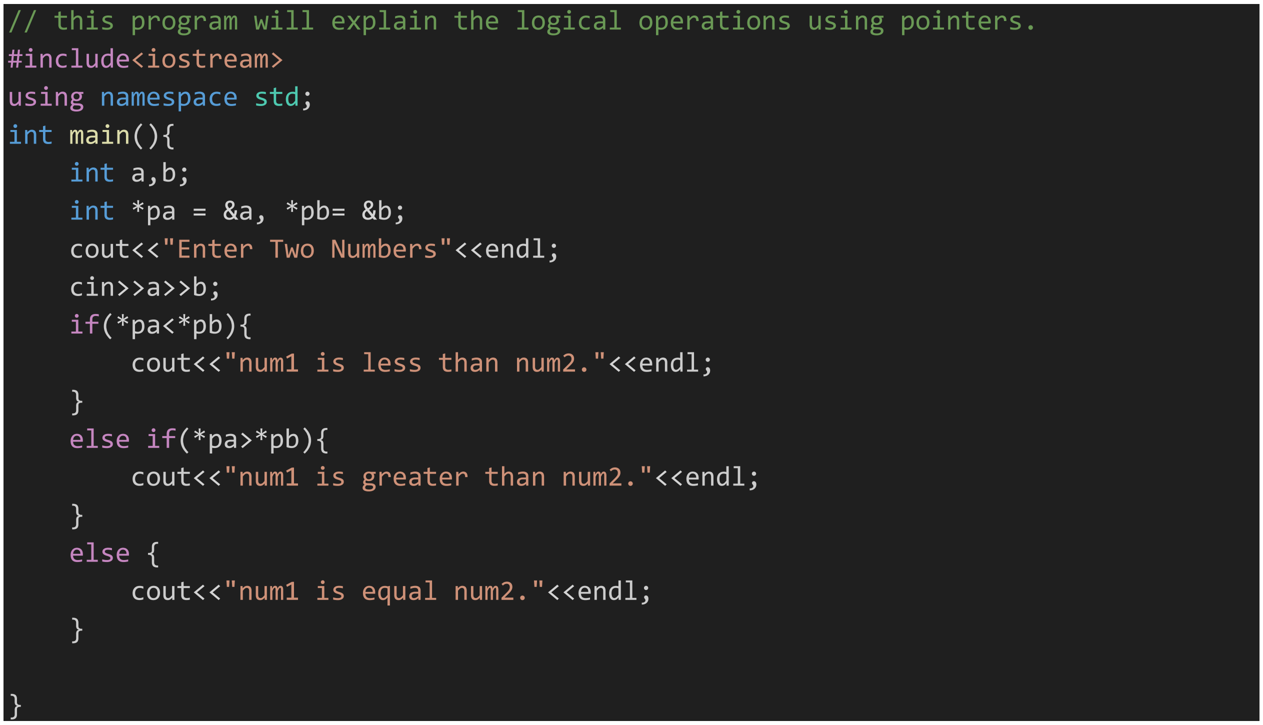


```
// this program explains the arithmatic operation on pointers.
#include<iostream>
using namespace std;
int main(){
    int n1=3,n2=2,sum,sub,mul,dev,mod;
    sum=sub=mul=dev=mod=0;

    int *pn1 = &n1,
        *pn2 = &n2,
        *psum = &sum,
        *psub = &sub,
        *pmul = &mul,
        *pdev = &dev,
        *pmod = &mod;

    *psum = *pn1 + *pn2;
    *psub = *pn1 - *pn2;
    *pmul = *pn1 * (*pn2);
    *pdev = *pn1 / *pn2;
    *pmod = *pn1 % *pn2;
    cout<<"Sum: value at Pointer: "<< *psum<<"\tValue at variable "<<sum<<endl;
    cout<<"Sub: value at Pointer: "<< *psub<<"\tValue at variable "<<sub<<endl;
    cout<<"Mul: value at Pointer: "<< *pmul<<"\tValue at variable "<<mul<<endl;
    cout<<"Div: value at Pointer: "<< *pdev<<"\tValue at variable "<<dev<<endl;
    cout<<"Mod: value at Pointer: "<< *pmod<<"\tValue at variable "<<mod<<endl;

}
```

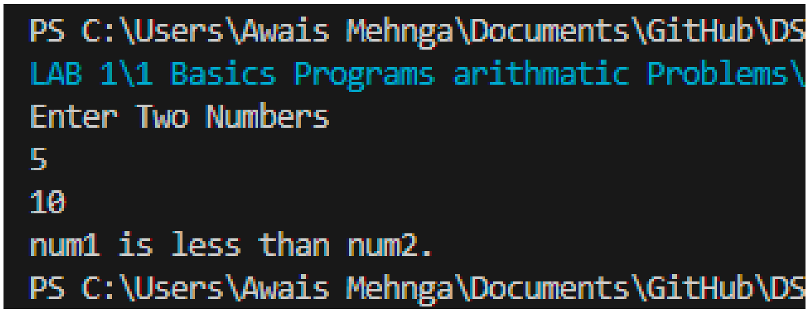


```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Documents\GitHub\DSA_
" ; if ($?) { g++ 1Arithmatic.cpp -o 1Arithmatic } ; if ($?) { .\1Arithmatic }
Sum: value at Pointer: 5        Value at variable 5
Sub: value at Pointer: 1        Value at variable 1
Mul: value at Pointer: 6        Value at variable 6
Div: value at Pointer: 1        Value at variable 1
Mod: value at Pointer: 1        Value at variable 1
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\1 Basics Programs arithmatic Problems> 
```

Program 2

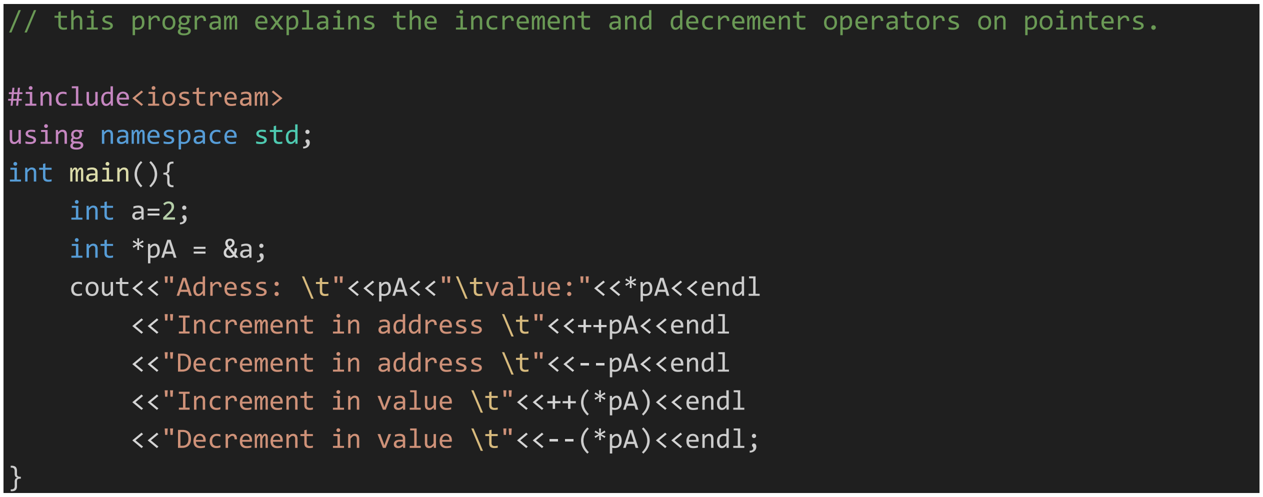


```
// this program will explain the logical operations using pointers.
#include<iostream>
using namespace std;
int main(){
    int a,b;
    int *pa = &a, *pb= &b;
    cout<<"Enter Two Numbers"<<endl;
    cin>>a>>b;
    if(*pa<*pb){
        cout<<"num1 is less than num2."<<endl;
    }
    else if(*pa>*pb){
        cout<<"num1 is greater than num2."<<endl;
    }
    else {
        cout<<"num1 is equal num2."<<endl;
    }

}
```

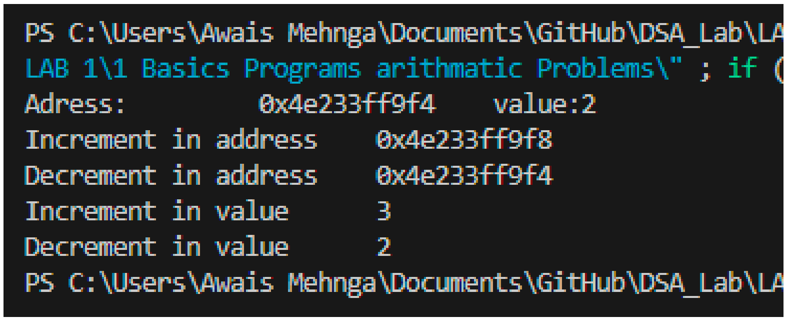


```
PS C:\Users\Awais Mehnga\Documents\GitHub\DS
LAB 1\1 Basics Programs arithmatic Problems\
Enter Two Numbers
5
10
num1 is less than num2.
PS C:\Users\Awais Mehnga\Documents\GitHub\DS
```

Program 3

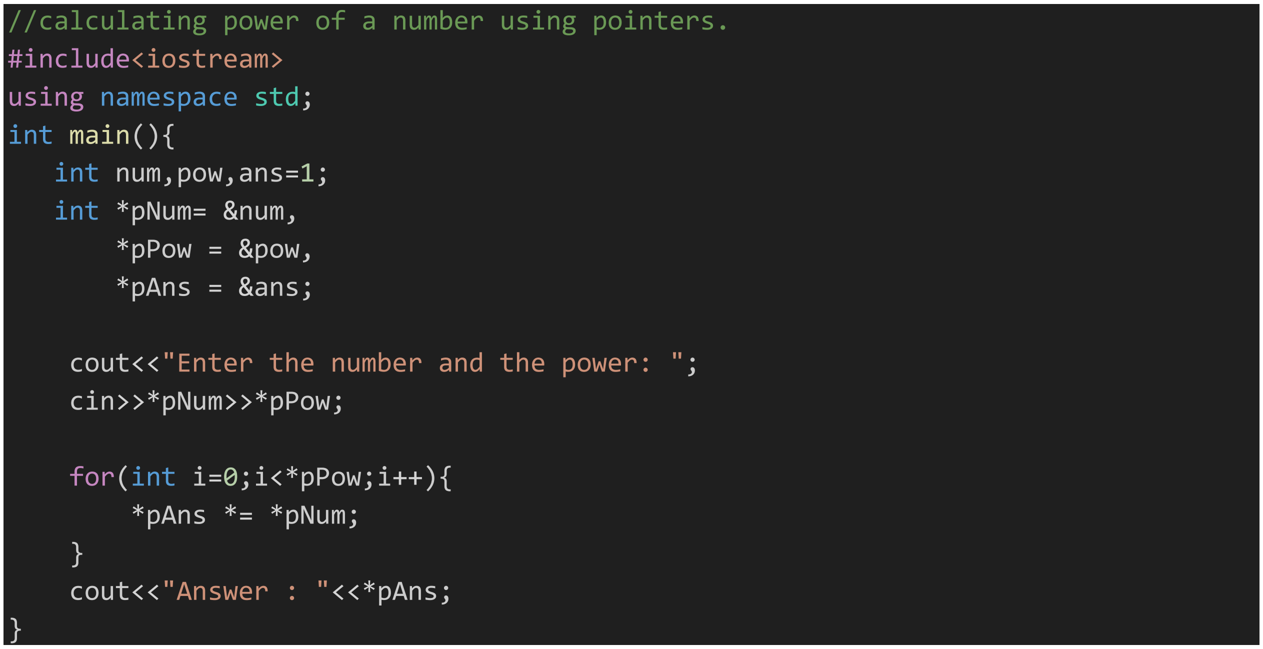


```
// this program explains the increment and decrement operators on pointers.

#include<iostream>
using namespace std;
int main(){
    int a=2;
    int *pA = &a;
    cout<<"Adress: \t"<<pA<<"\tvalue:"<<*pA<<endl
        <<"Increment in address \t"<<++pA<<endl
        <<"Decrement in address \t"<<--pA<<endl
        <<"Increment in value \t"<<++(*pA)<<endl
        <<"Decrement in value \t"<<--(*pA)<<endl;
}
```

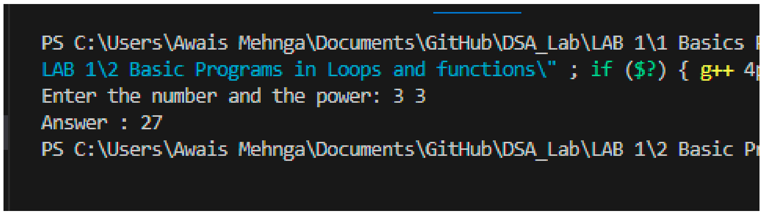


```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LA
LAB 1\1 Basics Programs arithmatic Problems\" ; if (
Adress:         0x4e233ff9f4    value:2
Increment in address    0x4e233ff9f8
Decrement in address    0x4e233ff9f4
Increment in value      3
Decrement in value      2
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LA
```

Program 4

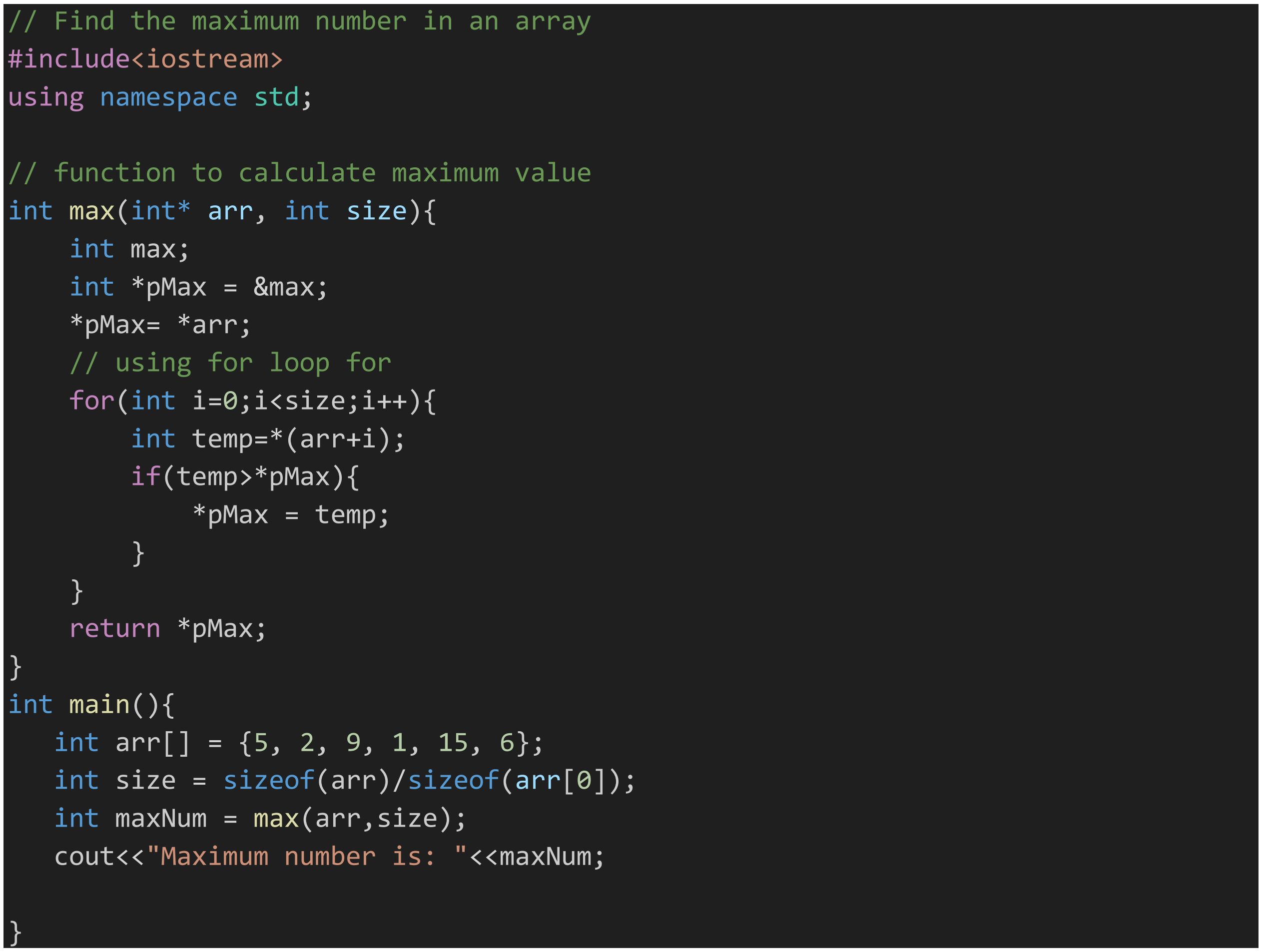


```
//calculating power of a number using pointers.
#include<iostream>
using namespace std;
int main(){
   int num,pow,ans=1;
   int *pNum= &num,
       *pPow = &pow,
       *pAns = &ans;

    cout<<"Enter the number and the power: ";
    cin>>*pNum>>*pPow;

    for(int i=0;i<*pPow;i++){
        *pAns *= *pNum;
    }
    cout<<"Answer : "<<*pAns;
}
```

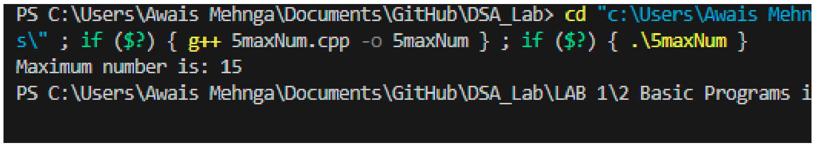


```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\1 Basics 
LAB 1\2 Basic Programs in Loops and functions\" ; if ($?) { g++ 4
Enter the number and the power: 3 3
Answer : 27
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic P
```

Program 5

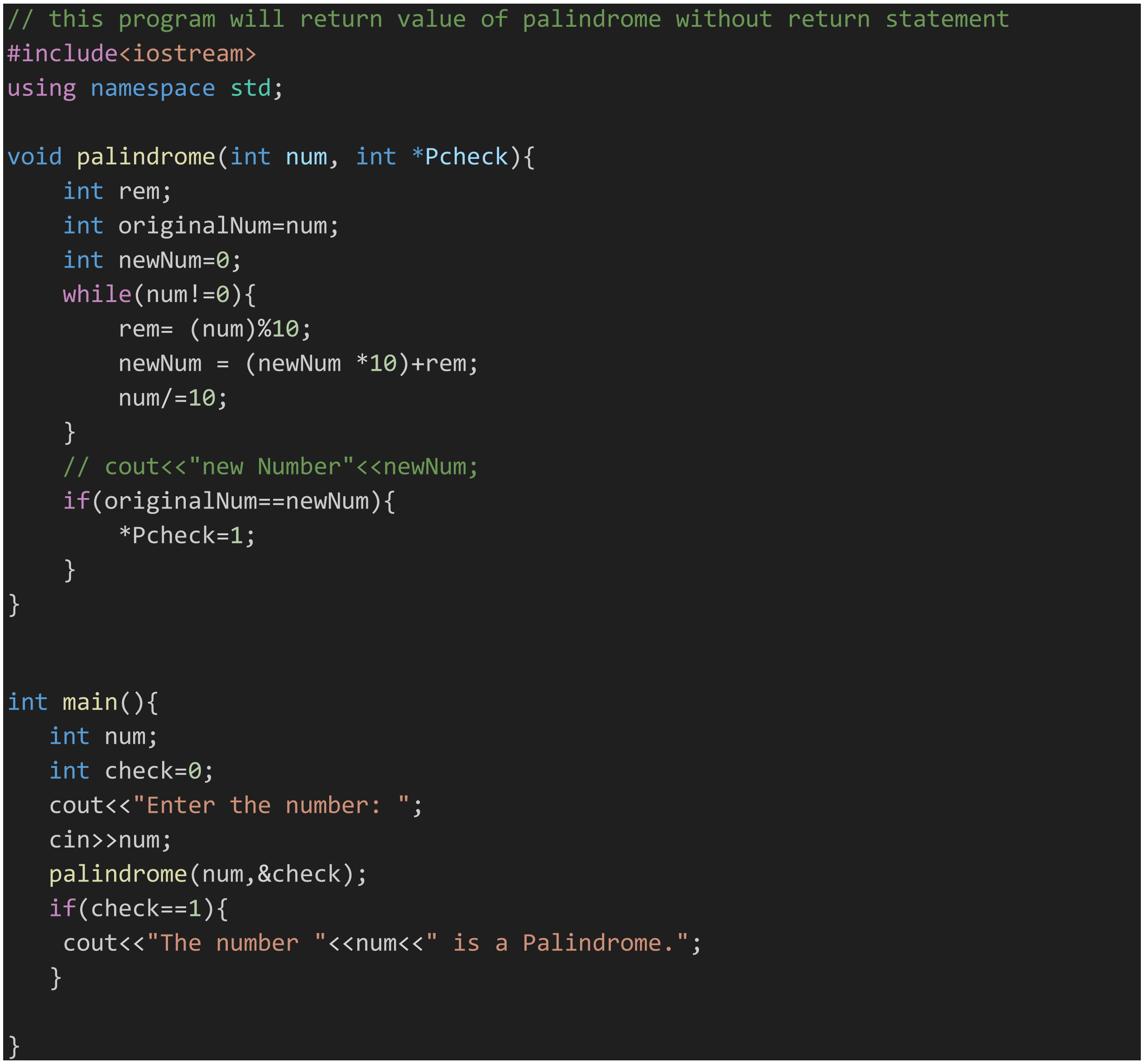


```
// Find the maximum number in an array
#include<iostream>
using namespace std;

// function to calculate maximum value
int max(int* arr, int size){
    int max;
    int *pMax = &max;
    *pMax= *arr;
    // using for loop for
    for(int i=0;i<size;i++){
        int temp=*(arr+i);
        if(temp>*pMax){
            *pMax = temp;
        }
    }
    return *pMax;
}
int main(){
   int arr[] = {5, 2, 9, 1, 15, 6};
   int size = sizeof(arr)/sizeof(arr[0]);
   int maxNum = max(arr,size);
   cout<<"Maximum number is: "<<maxNum;

}
```

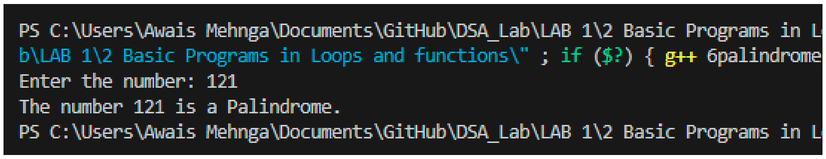


```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehn
s\" ; if ($?) { g++ 5maxNum.cpp -o 5maxNum } ; if ($?) { .\5maxNum }
Maximum number is: 15
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs i
```

Program 6



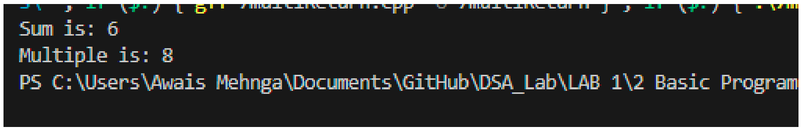
```cpp
// this program will return value of palindrome without return statement
#include<iostream>
using namespace std;

void palindrome(int num, int *Pcheck){
    int rem;
    int originalNum=num;
    int newNum=0;
    while(num!=0){
        rem= (num)%10;
        newNum = (newNum *10)+rem;
        num/=10;
    }
    // cout<<"new Number"<<newNum;
    if(originalNum==newNum){
        *Pcheck=1;
    }
}


int main(){
   int num;
   int check=0;
   cout<<"Enter the number: ";
   cin>>num;
   palindrome(num,&check);
   if(check==1){
    cout<<"The number "<<num<<" is a Palindrome.";
   }

}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs in L
b\LAB 1\2 Basic Programs in Loops and functions\" ; if ($?) { g++ 6palindrome
Enter the number: 121
The number 121 is a Palindrome.
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs in L
```

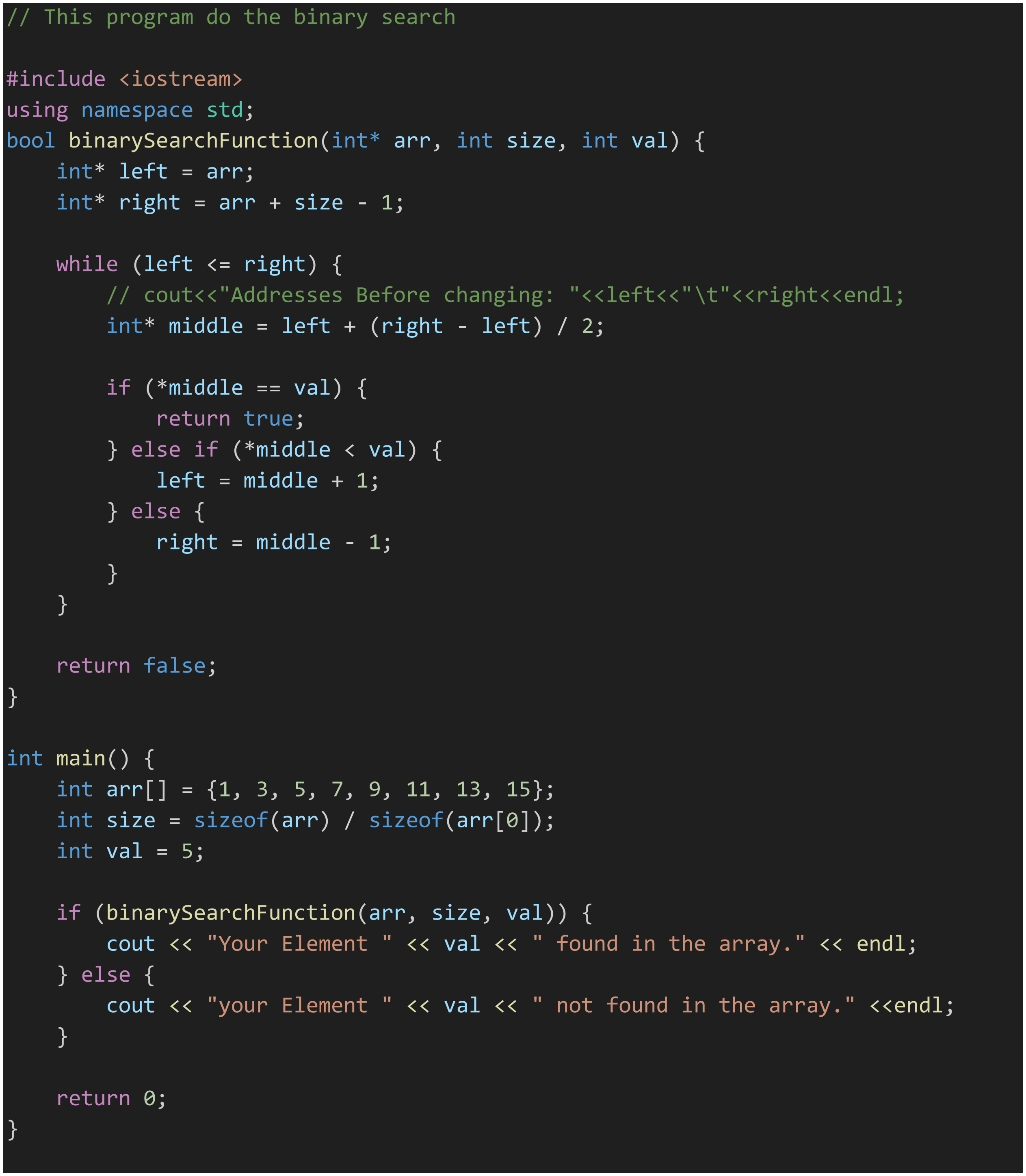
Program 7

```cpp
// as a function can only return single value but uising pointer web can
illusionate to return multiple values
#include<iostream>
using namespace std;

void returnMultiple(int a,int b,int *sum,int *mul){
    *sum = a+b;
    *mul = a*b;
}

int main(){
   int a,b,sum,mul;
   a=2;
   b=4;
   returnMultiple(a,b,&sum,&mul);
   cout<<"Sum is: "<<sum<<endl
       <<"Multiple is: "<<mul;
}
```

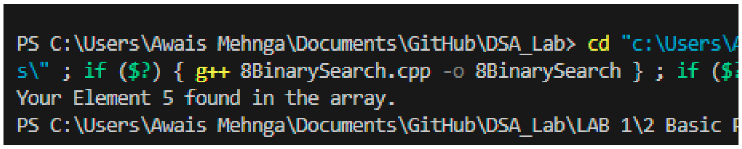


```
; if ($?) { g++ .\MultiReturn.cpp -o .\MultiReturn } ; if ($?) { .\Mu
Sum is: 6
Multiple is: 8
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Program
```

Program 8



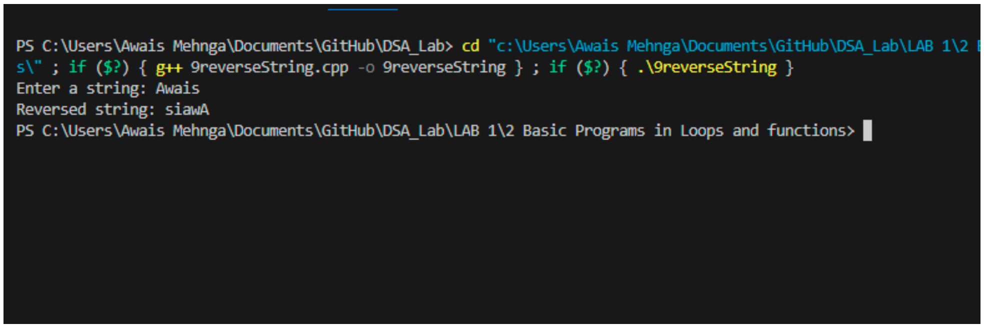
```
// This program do the binary search

#include <iostream>
using namespace std;
bool binarySearchFunction(int* arr, int size, int val) {
    int* left = arr;
    int* right = arr + size - 1;

    while (left <= right) {
        // cout<<"Addresses Before changing: "<<left<<"\t"<<right<<endl;
        int* middle = left + (right - left) / 2;

        if (*middle == val) {
            return true;
        } else if (*middle < val) {
            left = middle + 1;
        } else {
            right = middle - 1;
        }
    }

    return false;
}

int main() {
    int arr[] = {1, 3, 5, 7, 9, 11, 13, 15};
    int size = sizeof(arr) / sizeof(arr[0]);
    int val = 5;

    if (binarySearchFunction(arr, size, val)) {
        cout << "Your Element " << val << " found in the array." << endl;
    } else {
        cout << "your Element " << val << " not found in the array." <<endl;
    }

    return 0;
}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\A
s\" ; if ($?) { g++ 8BinarySearch.cpp -o 8BinarySearch } ; if ($
Your Element 5 found in the array.
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic P
```

Program 9

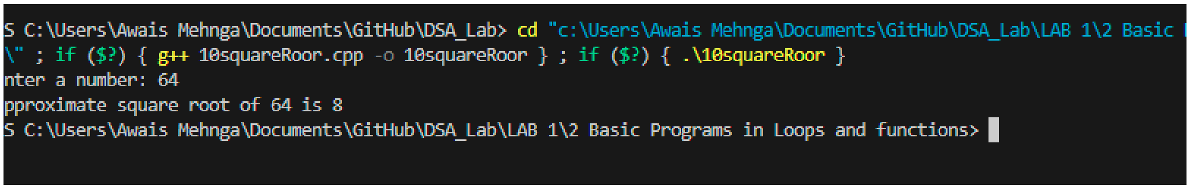
```
#include <iostream>
#include <cstring>
using namespace std;

void reverseString(char* str) {
    if (str == nullptr) {
        return; // Handle null pointer here
    }

    char* start = str;
    char* end = str + strlen(str) - 1;

/****Note*****/
/*the pointer to character array don't give addresses as the string in c/c++ are
null terminating strings
when the pointer moves to next memory location it by default cout the characters
from that location to the
null character location. that's why in this program the start and end are giving
the characters not the addresses.
However in previous binary search program the left and right pointers were giving
the addresses.
 */
    while (start < end) {

        // Swap the characters pointed to by start and end
        char temp = *start;
        *start = *end;
        *end = temp;
        // here moving the adress of the vars of string
        start++;
        end--;

    }
}

int main() {
    const int maxLength = 100;
    char inputString[maxLength];

    cout << "Enter a string: ";
    cin.getline(inputString, maxLength);

    // Reverse the input string
    reverseString(inputString);
    cout << "Reversed string: " << inputString << endl;

    return 0;
}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 B
s\" ; if ($?) { g++ 9reverseString.cpp -o 9reverseString } ; if ($?) { .\9reverseString }
Enter a string: Awais
Reversed string: siawA
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs in Loops and functions>
```

Program 10

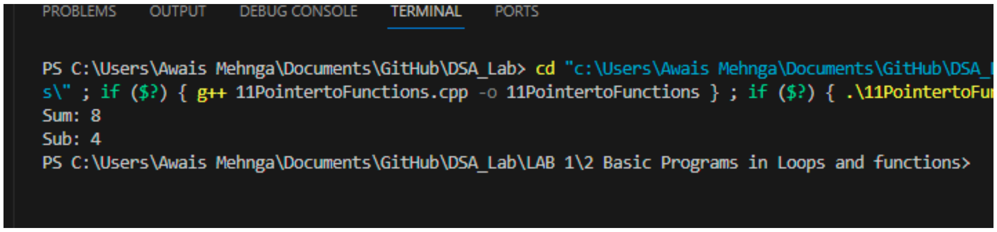
```
// Program to calculate the calculate the square root of a number.
#include <iostream>
#include <cmath>

using namespace std;

// Function to calculate square root using the Newton-Raphson method with
pointers
double calculateSquareRoot(double *number, double epsilon = 0.00001) {
    double approximation = (*number) / 2.0;
    double* ptrApproximation = &approximation;

    while (true) {
        double newApproximation = 0.5 * (*ptrApproximation + (*number) /
(*ptrApproximation));
        // cout<<"approx: "<<newApproximation<<" differnce:
"<<fabs(newApproximation - *ptrApproximation)<<endl;
        if (fabs(newApproximation - *ptrApproximation) < epsilon) {
            return newApproximation;
        }

        *ptrApproximation = newApproximation;
    }
}

int main() {
    double number;
    cout << "Enter a number: ";
    cin >> number;

    if (number < 0) {
        cout << "Square root is not defined for negative numbers." << endl;
    } else {
        double result = calculateSquareRoot(&number);
        cout << "Approximate square root of " << number << " is " << result <<
endl;
    }

    return 0;
}
```



```
S C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic 
\" ; if ($?) { g++ 10squareRoor.cpp -o 10squareRoor } ; if ($?) { .\10squareRoor }
nter a number: 64
pproximate square root of 64 is 8
S C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs in Loops and functions> 
```

Program 11

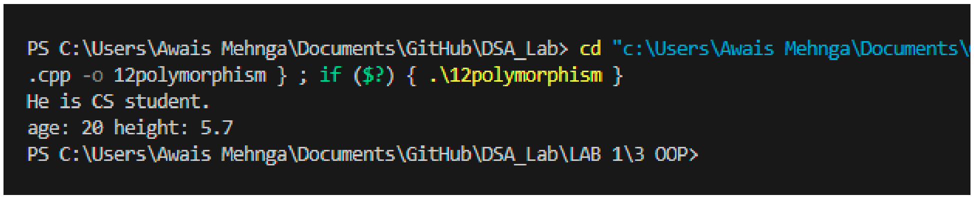
```
// this programs explains the pointers to a function.

#include<iostream>
using namespace std;

void add(int a,int b){
    cout<<"Sum: "<<a+b<<endl;
}
void sub(int a,int b){
    cout<<"Sub: "<<a-b<<endl;
}

int main(){
    char op;
    void (*ptrFunction)(int,int);
    int a=6,b=2;
    ptrFunction=add;
    ptrFunction(a,b);
    ptrFunction=sub;
    ptrFunction(a,b);

}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Documents\GitHub\DSA_
s\" ; if ($?) { g++ 11PointertoFunctions.cpp -o 11PointertoFunctions } ; if ($?) { .\11PointertoFu
Sum: 8
Sub: 4
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\2 Basic Programs in Loops and functions>
```

Program 12

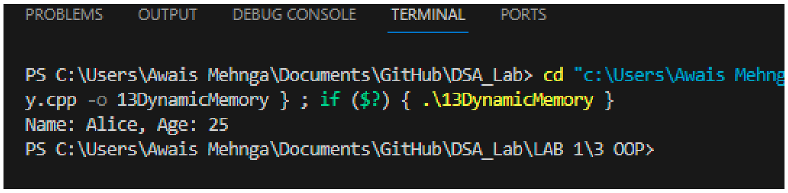
```
// basic use of pointer in oop
#include<iostream>
using namespace std;

class Awais{
    public:
        virtual void about(){ //virtual function allows polymorphism
            cout<<"He is CS student."<<endl;
        }
};
class Data: public Awais{
    private:
        int age=20;
        float height=5.7;
    public:
        void about() override{
            cout<<"age: "<<age<<" height: "<<height<<endl;
        }


};

int main(){
    Awais a; //simple object
    Awais* data; //this pointer points to the base class.
    Data aboutData;
    data = &aboutData; // this give referece of Data object to main class pointer.

    a.about(); //this simple object calls the base class function
    data->about(); // this pointer object to Data class calls the about of base class to which it was referenced
}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Documents\
.cpp -o 12polymorphism } ; if ($?) { .\12polymorphism }
He is CS student.
age: 20 height: 5.7
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\3 OOP>
```

Program 13

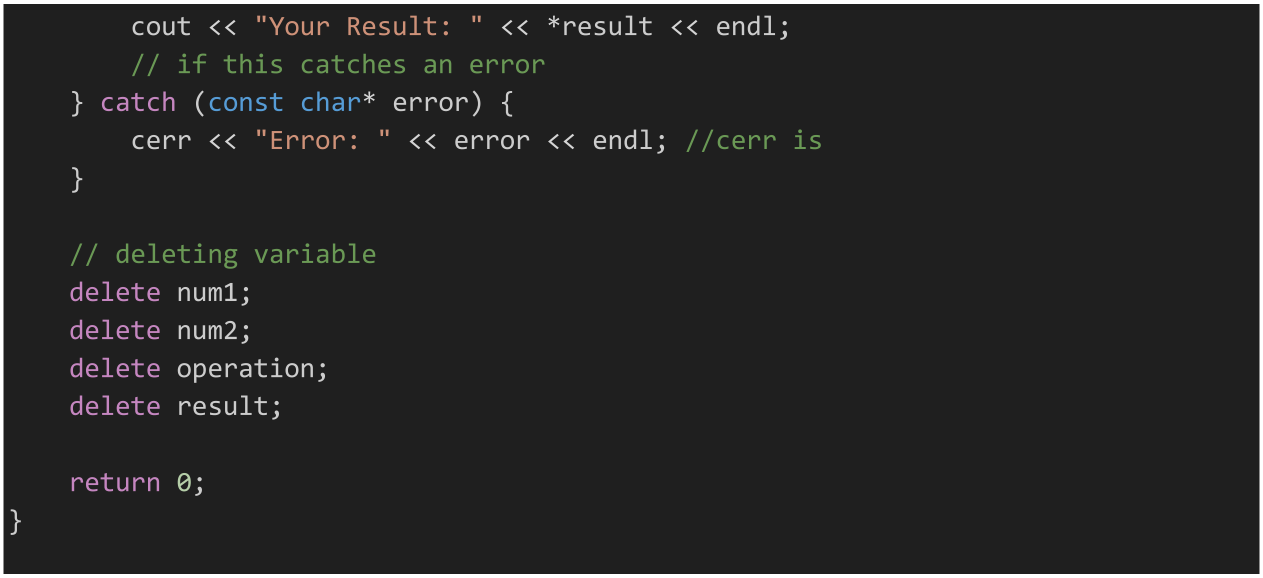
```
// this programs shows the dynamic memory allocation using pointers
#include <iostream>
using namespace std;

class Person {
public:
    Person(string name, int age) : name_(name), age_(age) {}

    void DisplayInfo() {
        cout << "Name: " << name_ << ", Age: " << age_ << endl;
    }

private:
    string name_;
    int age_;
};

int main() {
    // Dynamically allocate memory for a Person object
    Person* personPtr = new Person("Alice", 25);
    personPtr->DisplayInfo();

    // Deallocate the memory to prevent memory leaks
    delete personPtr;

    return 0;
}
```



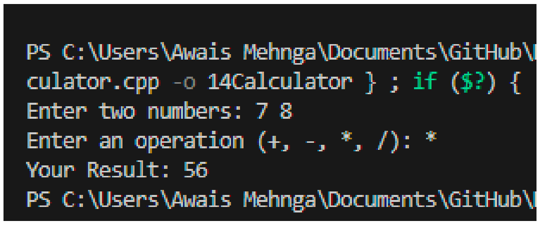
```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehng
y.cpp -o 13DynamicMemory } ; if ($?) { .\13DynamicMemory }
Name: Alice, Age: 25
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\3 OOP>
```

Program 14

```
// calculator application using pointers
#include <iostream>
using namespace std;

int main() {
    int *num1 = new int;
    int *num2 = new int;
    char *operation = new char;
    int *result = new int;

    cout << "Enter two numbers: ";
    cin >> *num1 >> *num2;

    cout << "Enter an operation (+, -, *, /): ";
    cin >> *operation;

    try {
        if (*operation == '+') {
            *result = *num1 + *num2;
        } else if (*operation == '-') {
            *result = *num1 - *num2;
        } else if (*operation == '*') {
            *result = *num1 * *num2;
        } else if (*operation == '/') {
            if (*num2 == 0) {
                throw "Division by zero is not allowed.";
            }
            *result = *num1 / *num2;
        } else {
            throw "Invalid operation.";
        }
```

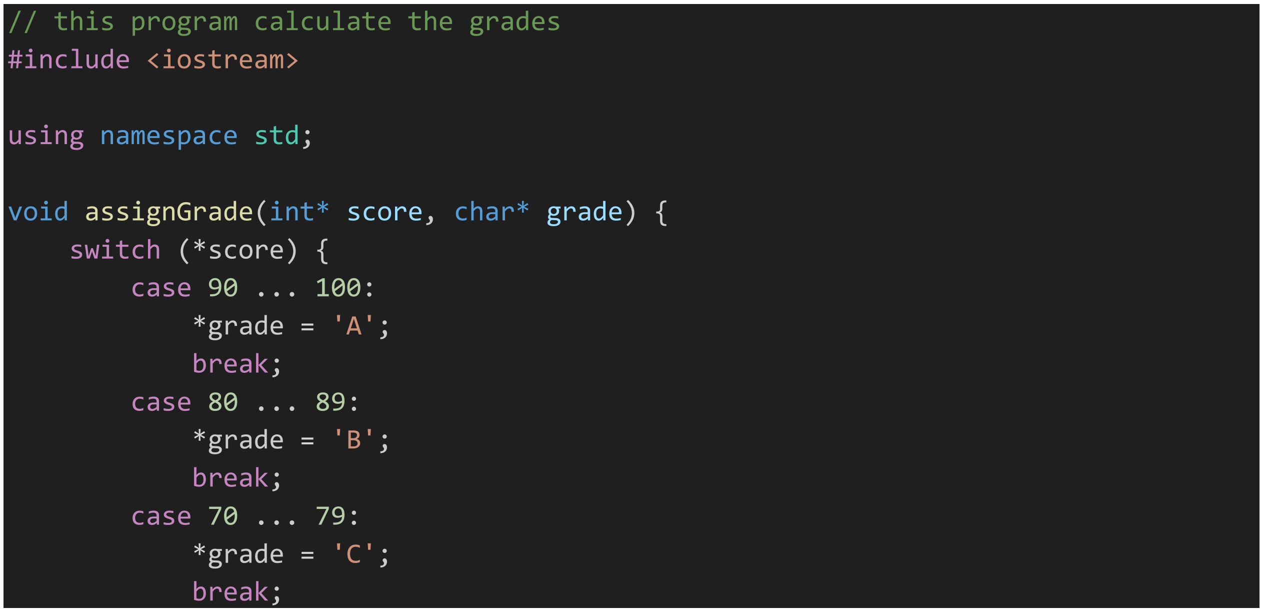


```
        cout << "Your Result: " << *result << endl;
        // if this catches an error
    } catch (const char* error) {
        cerr << "Error: " << error << endl; //cerr is
    }

    // deleting variable
    delete num1;
    delete num2;
    delete operation;
    delete result;

    return 0;
}
```

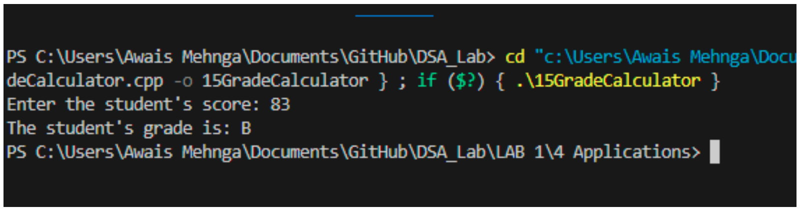


```
PS C:\Users\Awais Mehnga\Documents\GitHub\
culator.cpp -o 14Calculator } ; if ($?) {
Enter two numbers: 7 8
Enter an operation (+, -, *, /): *
Your Result: 56
PS C:\Users\Awais Mehnga\Documents\GitHub\
```

Program 15



```
// this program calculate the grades
#include <iostream>

using namespace std;

void assignGrade(int* score, char* grade) {
    switch (*score) {
        case 90 ... 100:
            *grade = 'A';
            break;
        case 80 ... 89:
            *grade = 'B';
            break;
        case 70 ... 79:
            *grade = 'C';
            break;
        case 60 ... 69:
            *grade = 'D';
            break;
        default:
            *grade = 'F';
            break;
    }
}

int main() {
    int studentScore;
    char studentGrade;

    cout << "Enter the student's score: ";
    cin >> studentScore;

    assignGrade(&studentScore, &studentGrade);

    cout << "The student's grade is: " << studentGrade << endl;

    return 0;
}
```



```
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab> cd "c:\Users\Awais Mehnga\Docu
deCalculator.cpp -o 15GradeCalculator } ; if ($?) { .\15GradeCalculator }
Enter the student's score: 83
The student's grade is: B
PS C:\Users\Awais Mehnga\Documents\GitHub\DSA_Lab\LAB 1\4 Applications> 
```