Index

Assignment No. 15-May-2021

Himanshu Gupta CSE49

Pull with Git?

This assignment is best viewed on GitHub. Use above link.

Index

Question No. 1 :

Write a function using reference variables as arguments to swap the values of a pair of integers.

Code

```cpp
//QUESTION Write a function using reference variables as arguments to swap the values of a pair of
↪  integers.

#include <iostream>

void swap(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}

int main()
{
    std::cout << "Enter two Numbers: " << "\n";
    int a, b;
    std::cin >> a >> b;
    std::cout << "Before Swapping: " << "\n" << "a: " << a << "\n" << "b: " << b << "\n";
    swap(a , b);
    std::cout << "After Swapping: " << "\n" << "a: " << a << "\n" << "b: " << b << "\n";
    return 0;
}
```

$ Assign 15-May-2021/bin/1.o

-

Input

```
18 90
```

Output

```
Enter two Numbers:
Before Swapping:
a: 18
b: 90
After Swapping:
a: 90
b: 18
```

Question No. 2 :

> Write a function that creates a vector of user given size M using new operator.

Code

```
//QUESTION Write a function that creates a vector of user given size M using new operator.

#include <iostream>
#include <vector>

int main()
{
    int M;
    std::cout << "Enter the size of the vector you want to create: ";
    std::cin >> M;
    auto vec = new std::vector<int>(M , 0);
    std::cout << "The size of the vector is: " << vec->size() << '\n';
    std::cout << "The Elements of the vector are: " << '\n';
    for(auto it : *vec)
        std::cout << it << '\n';
    return 0;
}
```

$ Assign 15-May-2021/bin/2.o

-

Input

```
23
```

Output

```
Enter the size of the vector you want to create: The size of the vector is: 23
The Elements of the vector are:
0
0
0
...
```

Question No. 3 :

> Write a program to evaluate the following investment equation

Code

```
//QUESTION Write a program to evaluate the following investment equation
//V = P(1+r)n
//and print the tables which would give the value of V for various
//of the following values of P, r and n:
//P: 1000, 2000, 3000,……………,10,000
//r: 0.10, 0.11, 0.12,…………………….,0.20
//n: 1, 2, 3,………………………………..,10

#include <iostream>

int main()
{
    std::cout << "|P    "
              << "|r    "
              << "|n    "
              << "|V|" << std::endl;
    std::cout << "|---|---|---|---|" << std::endl;
    for (int n = 1; n <= 10; n++)
    {
        for (int r = 10; r <= 20; r++) // Don't work with doubles use ints and then divide by 10 later.
        {
            for (int P = 10; P <= 100; P += 10)
            {
                long V = P * (100 + r) * n;
                std::cout << '|' << P*10 << "    |" << r/double(100) << "    |" << n << "    |" << V <<
 ↪  '|' << "\n";
            }
        }
    }
}
```

$ Assign 15-May-2021/bin/3.o

-

Input

Output

```
|P |r |n |V|
|---|---|---|---|
|100 |0.1 |1 |1100|
|200 |0.1 |1 |2200|
|300 |0.1 |1 |3300|
|400 |0.1 |1 |4400|
...
```

Question No. 4 :

An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the vote cast for each candidate using an array variable count. In case, a number read is outside the range 1 to 5, the ballot should be considered as a "spoilt ballot" and the program should also count the numbers of "spoilt ballots".

Code

```
//QUESTION An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting
↪  is done by marking the candidate number on the ballot paper. Write a program to read the ballots and
↪  count the vote cast for each candidate using an array variable count. In case, a number read is
↪  outside the range 1 to 5, the ballot should be considered as a "spoilt ballot" and the program should
↪  also count the numbers of "spoilt ballots".

#include <iostream>
#include <array>

int main()
{
    std::array<int , 5> count ;
    count.fill(0);
    int spoiltVotes = 0;
    std::cout << "Enter who to vote(from 5 candidates): " << std::endl;
    for(int i; std::cin >> i; )
    {
        if(i > 5 || i < 1)
            spoiltVotes++;
        else
            count[i-1]++;
    }
    std::cout << "No of Votes: " << std::endl;
    std::cout << "|Candidate|Ballot Count|" << std::endl;
    std::cout << "|---|---|" << std::endl;
    for(int it = 0; it < count.size(); it++)
    {
        std::cout << "|" <<  it+1 << "|   " << count[it] << "|\n";
    }
    std::cout << "Total No. of spoilt votes: " << spoiltVotes;
}
```

$ Assign 15-May-2021/bin/4.o

-

Input

```
7
3
5
4
7
1
3
7
4
...
```

Output

```
Enter who to vote(from 5 candidates):
No of Votes:
|Candidate|Ballot Count|
|---|---|
|1| 16|
|2| 14|
|3| 11|
|4| 16|
|5| 11|
Total No. of spoilt votes: 32
```

Question No. 5 :

Write a program to evaluate the following function to 0.0001% accuracy

Code

```cpp
//QUESTION Write a program to evaluate the following function to 0.0001% accuracy
/*
 *(a) sin x = x - x3/3! + x5/5! - x7/7! +...........
 *(b) SUM = 1+ (1/2)2 + (1/3)3 + (1/4)4 + .........
 *(c) Cos x = 1 -x2/2! + x4/4! - x6/6! + .........
*/

#include <iostream>
#include <cmath>

namespace mine
{
    long factorial(int n)
    {
        if (n <= 1)
            return 1;
        return n * factorial(n - 1);
    }

    double sin(double angle, int stage = 0)
    {
        double el = (pow(-1, stage) * pow(angle, 2 * stage + 1)) / factorial(2 * stage + 1);
        if (fabs(el) <= 0.000001 && stage != 0)
            return 0;
        return el + sin(angle, stage + 1);
    }
    double cos(double angle, int stage = 0)
    {
        double el = (pow(-1, stage) * pow(angle, 2 * stage)) / factorial(2 * stage);
        if (fabs(el) <= 0.000001 && stage != 0)
            return 0;
        return el + cos(angle, stage + 1);
    }
    constexpr inline double sum(int stage = 1)
    {
        double el = pow(1 / stage, stage);
        if (fabs(el) <= 0.000001)
            return 0;
        return el + sum(stage + 1);
    }
}

int main()
{
    double angleFsin = 3, angleFcos = 2;
    std::cout << "Enter angle for sin and then cosine to calculate: \n";
    std::cin >> angleFsin >> angleFcos;
    std::cout << "Sin: " << mine::sin(angleFsin) << std::endl
              << "Cos: " << mine::cos(angleFcos) << std::endl
              << "The sum that does not take a argument: " << mine::sum();
    return 0;
}
```

$ Assign 15-May-2021/bin/5.o

-

Input

Output

```
12
13
14
15
16
17
18
Enter angle for sin and then cosine to calculate:
Sin: 0.14112
Cos: -0.416147
The sum that does not take a argument: 1
```

Assignment No. 25-May-2021

Himanshu Gupta CSE49

Pull with Git?

> This assignment is best viewed on GitHub. Use above link.

Index

Question No. 1 :

> Write and run a program that directly implements the quotient operator / and the remainder operator % for the division of positive integers.

Code

```cpp
//QUESTION Write and run a program that directly implements the quotient operator / and the remainder
↪ operator % for the division of positive integers.

#include <iostream>
#include <utility>

struct DivQues : public std::pair<int, int> //inheritied class DivQues for storage of divident and divisor
{
    using std::pair<int , int>::pair; //inheriting constructor from std::pair
};
std::ostream &operator<<(std::ostream &os, DivQues const &l_div_result) //ostream insertion overload for
↪ DivQues, prints in a pretty format.
{
    os << "Divident: " << l_div_result.first << '\t'
       << "Divisor: " << l_div_result.second;
    return os;
}
std::istream &operator>>(std::istream &is, DivQues &divis) //istream extraction overload for DivQues, puts
↪ 2 ints in DivQues
```

```
{
    std::string dividend, divisor;
    is >> dividend;
    if(is.eof())    //checks for eof after every extraction
        return is;
    is >> divisor;
    if(is.eof())
        return is;
    DivQues divisTemp({std::stoi(dividend), std::stoi(divisor)}); //exception safety, if throws divis is
 ↪  still valid and we are rid of bad input
    divis = std::move(divisTemp);
    return is;
}

struct DivAns : public std::pair<int, int> //inherited class DivAns for storage of quotient and remainder
{
    using std::pair<int , int>::pair;
};
std::ostream &operator<<(std::ostream &os, DivAns const &l_div_result)
{
    os << "Quotient: " << l_div_result.first << '\t'
       << "Remainder: " << l_div_result.second;
    return os;
}

DivAns l_div(DivQues const &divis) //division function
{
    if (divis.second == 0)
        throw std::runtime_error("division by zero not possible");
    DivAns result;
    result.first = divis.first / divis.second;
    result.second = divis.first % divis.second;
    return result;
}

int main()
{
    std::cout << "Keyboard Interupt to Exit" << std::endl;
    while (true)
    {
        DivQues divis;

        try
        {
            std::cin >> divis;
        }
        catch (const std::out_of_range &e)  //Detects out of range input
        {
            std::cerr << "Integer Boundary Error: out_of_range" << '\n';
            std::cout << std::endl;
            continue;
        }
        catch (const std::invalid_argument &e) //Detects invalid input
        {
            std::cerr << "Invalid Argument" << '\n';
            std::cout << std::endl;
            continue;
        }

        if(std::cin.eof()) //Breaks if EOF reached
            break;
        std::cout << divis << std::endl;

        try
        {
            std::cout << l_div(divis) << '\n';
        }

        catch (const std::runtime_error &e) //Detects if dividing by zero
        {
            std::cerr << "Exception Caught: " << e.what() << '\n';
        }
        std::cout << std::endl;

    }
}
```

$ Assign 25-May-2021/bin/1.o

-

Input
123 32
234 0
21 7
1223453145141234 323
12 3
23

Output

```
Keyboard Interupt to Exit
Divident: 123 Divisor: 32
Quotient: 3 Remainder: 27

Divident: 234 Divisor: 0
Exception Caught: division by zero not possible

Divident: 21 Divisor: 7
Quotient: 3 Remainder: 0

Integer Boundary Error: out_of_range

Divident: 12 Divisor: 3
Quotient: 4 Remainder: 0
```

Question No. 2 :

Write a function which evaluates the third degree polynomial a0 + a1x + a2x2 + a3x3. Also implement this function using Horner's Algorithm, grouping the calculations as a0 + (al + (a2 + a3x)x)x . Compare the efficiency of both the functions.

Code

```
//QUESTION Write a function which evaluates the third degree polynomial a0 + a1x + a2x2 + a3x3. Also
↪  implement this function using Horner's Algorithm, grouping the calculations as a0 + (al + (a2 +
↪  a3x)x)x . Compare the efficiency of both the functions.

long horner_impl_poly_three(long x , long a0 = 1 , long a1 = 2, long a2 = 43, long a3 = 3)
{
    /* Average time (cacluating 100 billion instances with x from 0 ... 100billion) from 1000 runs using
     ↪  GNU time and GNU parralel
     * took net 4min 21secs to run 1000 jobs in parallel on a 8 core processor with SMT clocked at 4.0ghz
     * real 4.14
     * user 4.04
     * sys 0.00/0.004
    */
    return a0 + (a1 + (a2 + a3*x)*x)*x;
}

long poly_three(long x , long a0 = 1, long a1 = 2, long a2 = 43, long a3 = 3)
{
    /* Average time (cacluating 100 billion instances with x from 0 ... 100billion) from 1000 runs using
     ↪  GNU time and GNU parralel
     * took net 4min 54secs to run 1000 jobs in parallel on a 8 core processor with SMT clocked at 4.0ghz
     * real 4.67
     * user 4.58
     * sys 0.00/0.004
    */
    return a0 + a1*x + a2*x*x + a3*x*x*x;
}

int main()
{
    for(long x = 0 ; x < 1000000000; x++)
    {
        poly_three(x , 1 , 2 , 43 , 3);
    }
    /*
    for(long x = 0 ; x < 1000000000; x++)
    {
        horner_impl_poly_three(x , 1 , 2 , 43 , 3);
    }
    */
}
```

$ Assign 25-May-2021/bin/2.o

-

Input

Output

A note about time complexity:

```
It seems that time complexity of both algorithms are of O(1). This was verified using various
inputs of 'a' and 'x' which more or less ran in the same time. Along with that it also seems
that Horner's algorithm is a little bit faster. This may be due to less multiplication calls
in Horner's (3 compared to 6). But modern x86-64 processors are really fast at 64-bit int
multiplication, so the change only becomes apparent when the iteration count goes into the range
of tens of billions. It is also of note that the method used for calculating the average time was
multithreaded and was done with a used computer which could render the results, hence the average
erroneous. The sample of 1000 is also not that big.
```