

Functions Homework

For this homework your task was to write a set of functions to perform a series of tasks defined in `problems.h`. For each problem you needed to write a function which completes a task and then call the function from the main function in `problemX.cc`. Notice that all the function definitions are written in `problems.cc`. If you do not understand what function definitions and function declarations please read .

Some general notes on functions

If you need some clarification on functions I suggest you read some of the following links. Try writing the code yourself. Make sure you understand what functions, how to write a function, and how to call a function. This is a very important concept in programming.

Function Tutorial Basic Function

Problem 1

The task is to write a function named `print` which takes in a string and a vector and prints the elements of the vector onto the screen.

Declaration

All function declarations are in the header file `problems.h`

In the file “`problems.h`” we create the function declaration.

The function declaration tells the compiler three things: - The function name - The function return type. - The function inputs and input types.

We need to write a function that prints a **vector** of **ints** to the screen along with a label defined as a **string**. Since this function only prints values to a screen we know that the functions return type is `void`.

We declare the function in `problems.h` as so.

```
void print(const std::vector<int>& v, const string& label);
```

Definition

Now that we have declared the function we still have to provide a function definition. In this homework we place all the function definitions in `problems..` To define a function we need to write the exact function signature as in the declaration a and then provide the body of the code in the curly braces `{}`.

```
// function definition
void print(const vector<int>& v, const string& label) {
    cout<<label<<": "; // print out the label
    // v.size() gives us the size of the vector
```

```

    for (int i = 0; i < v.size(); i++){// for every value in the vector
        cout << v[i] << " ";// print v[i] followed by a space
    }
    cout<<endl;// finish by printing a new line after the last element
}

```

- Notice that the top line of the function definition matches the declaration.
- First we print the label
- We then use a for loop to access every element of the vector
 - For each element we print each element followed by a space
- We finish by printing a new line after the last element
- Notice that we do not return a value from this function

Now that we have declared the function this allows us to call the function from a main program or any other functions. For example,

```

void f(){
    vector<int> a={1, 2,3,4,5,};// create a vector of size 5
    vector<int> b(4);// create a vector of size 4 with default value 0
    print(a,"vector a");
    print(b,"vector b");
}

```

The code will print the vector of integers using functions you defined.

For problem 1 the main function is defined in `problem1.cc`

```

#include "problems.h"// this line is very important
// Here we are declaring the function

int main(){
    string label="my label";// create a label
    vector<int> v={1,2,3,4,6,9};// create a vector
    print(v,label);// print the vector with a label
}

```

Problem 2

Problem 2 is to write a function that creates a vector of Fibonacci numbers. The function should have 4 inputs Inputs: - `int x` the first number in the sequence - `int y` the second number in the sequence - `vector<int>& v` The vector that we are modifying to save the sequence - `int n`

Again the first task is to declare the function in `problems.h`. Since we are passing the vector that we are modifying as an input the return type will be `void`.

```

void fibonacci(int x,int y,vector<int>& v,int n);

```

Notice `&` that means we are passing the argument by reference. This basically means we are passing the actual object from the scope of where the function is being called from instead of just creating a copy of vector in the function scope. For those who are not familiar with pass by reference vs pass by value please read the following [link](#).

Here is the function definition. The task is to create a Fibonacci sequence of size `n`. The first step is to clear the vector that we pass in. It is possible the vector we pass in already has values therefore we clear it using `v.clear()`. `v.clear()` is a function that clears our vector. Since `x` and `y` are the first two values we can add those values into the first two locations using `v.push_back(x)` and `v.push_back(y)`.

To create the rest of the we need to add the rest of the values from position 2 all the way to position `n-1`. We do this with a for loop. For each value we are adding we simply push_back the sum of the previous two values.

```
void fibonacci(int x,int y,vector<int>&v,int n){
    v.clear();// clear vector`
    v.push_back(x);// push back first value [x]
    v.push_back(y);// [x,y]
    for(int i=2;i<n;i++){
        v.push_back(v[i-1]+v[i-2]);// push back the previous two
    }
}
```

Problem 3

Here we are using the fibonacci sequence we defined in the previous problem to find an approximation to the largest possible `int`. To find the solution first you should try to generate a large sequence of numbers using the fibonacci functions.

Fibonacci: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711

Process finished with exit code 0

If you look at the values you generated you will notice that the values increase until they eventually get too big and start becoming negative. The approximation for the largest `int` should be the value before the first value that becomes negative. We can find this number using a for loop.

Here is the main file `problem3.cc`

```
#include "problems.h"

int main(){
    string label="Fibonacci";
    vector<int> v={1,2,3,4,6,9};
    fibonacci(0,1,v,100);// call the fibonacci function
}
```

```

    print(v,label);

    int max=0;
    for(int i=0;i<v.size();i++){
        if(v[i]<0){//if v[i] is negative
            max=v[i-1];// the max value is the value before
            break;// we break to end the for loop
        }
    }
    cout<<"The largest int is "<<max<<endl;
}

```

After running the problem I get the following output.

The largest int is 1836311903

Problem 4

In this problem we need to write two functions that reverse a sequence. The first function passes a vector of ints by values and generates a new function. The second passes the original vector and reverses the values in place.

In problems.h we have the function definitions.

```

vector<int> reverse(vector<int>);
void reverse_swap(vector<int>& );

```

In problems.cc we have the definitions.

```

vector<int> reverse(vector<int> v){
    vector<int> rev_v;
    int n=v.size();// get the size of the vector
    // start at the back and go backwards to reverse
    // iterate till the front
    for(int i=n-1;i>=0;i--){
        rev_v.push_back(v[i]);
    }
    return rev_v;
}

void reverse_swap(vector<int>& v ){
    int n=v.size()-1;// get the last element of the vector
    for(int i=0;i<v.size()/2;i++){// only go to the middle element
        int tmp=v[i];//swap
        v[i]=v[n-i];// last element -1
        v[n-1]=tmp;//
    }
}

```

and here is the main function that tests the code

```
#include "problems.h"

int main(){
    string label="my label";
    vector<int> v={1,2,3,4,6,9};
    print(v,"original");
    vector<int> rev_v= reverse(v);
    reverse_swap(v);
    print(rev_v,"reverse(v)");
    print(v," reverse_swap(v)");
}
```

Problem 5

In problem 5 we need to write a function that takes two vectors of doubles. Representing the price and weight and computes the sum of all prices and weights. You need to make sure the sizes of the two vectors are the same.

Declaration

Since we know the function should return a single number that is the sum of all prices multiplied by weights we know the return type is `double`. Here is an example of a function declaration that works for this problem.

```
double productSum(vector<double> price,vector<double> weight);
```

Here is the definition

```
double productSum(vector<double> price,vector<double> weight){
    if(price.size()!=weight.size()){// check if sizes are correct
        return NAN;
    }
    double product_sum=0;// create initial value
    for(int i=0;i<=price.size();i++){
        product_sum+=price[i]*weight[i]; // sum = sum+ price*weight
    }
    return product_sum;
}
```

Here is a main function I used to test.

```
#include "problems.h"

int main(){
    string label="my label";
    vector<double> myPrice={1.0,2.4,3.23};
    vector<double> myWeights={2.0,4.4,5.0};
```

```

vector<double> myWeights_WrongSize={1.0,2.0,4.4,5.0};

double p_sum= productSum(myPrice,myWeights);
double p_sum2= productSum(myPrice,myWeights_WrongSize);

cout<<"product sum = "<<p_sum<<endl;
cout<<"product sum = "<<p_sum2<<endl;

}

```

Here is the output I generate.

```

product sum = 28.71
product sum = nan

```

Problem 6

In problem 6 we need to write a function to grab the largest value from a vector. As a good example I used the main function from problem2 to see if the max int approximation is in fact the largest value int. Here I just present the declaration, definition, and main function. ### Declaration

```
int maxv(vector<int>&v);
```

Definition

```

int maxv(vector<int>&v){

    int max=-100000;// set max to a small number
    for(int i =0;i<v.size();i++){
        if(v[i]>max){//if v[i] is greater than max then max is v[i]
            max=v[i];
        }
    }
    return max;
}

```

main function

```

#include "problems.h"

int main() {
    string label = "Fibonacci";
    vector<int> v = {1, 2, 3, 4, 6, 9};
    fibonacci(0, 1, v, 100);// call the fibonacci function
    int max = 0;
    for (int i = 0; i < v.size(); i++) {

```

```

        if (v[i] < 0) {//if v[i] is negative
            max = v[i - 1];// the max value is the value before
            break;// we break to end the for loop
        }
    }
    cout << "The largest int is " << max << endl;
    max = maxv(v);
    cout << "The largest int is " << max << endl;
}

```

Run and test this code

Problem 7

Write a function that finds the smallest and largest element of a vector argument. Also compute the mean and the median. Remember you can only return one variable from a function therefore you must pass the variables by reference to return all the variables. Try this problem on your own before you look up the solutions

You can find all the code for this homework here [Functions Lecture](#)