

## Assignment2

Before doing this assignment, you should already know the following things:

- Operator ++, -
- Using for loop and while loop
- Using if and switch
- Array
- How to use cout

## Review Questions

(You should write the answers to the following questions in the same file)

1. Write down the output of the program

```
int j = 0;
while (++j < 9) {
    cout << j++ << endl;
}
```

2. Write a loop that will output from 2 to the power of 1 to 2 to the power of 10. The output should look like: 2 4 8 16 32 64 128 256 512 1024
3. Read the following code. Rewrite it without **break** and **continue**

```
// you should run this program yourself first to understand it
int line = 0;
char ch;
// if there's any char data unread from the console, the loop will run, and each loop will read a char
// when character 'Q' is read, end the loop
while (cin.get(ch)) {
    if (ch == 'Q')
        break;
    if (ch != '\n')
        continue;
    line++;
}
```

Hint: your code should look like this:

```
while (conditions) {
    if (conditions) {
        line++;
    }
}
```

## Programming Exercise

In this exercise, you need to arrange an array of integers in ascending order using selection sort.

The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

For example: we have array  $arr = [5, 10, 6, 2, 3]$ , and integer  $i$  indicating the first  $i$  elements are sorted. -  $i=0$ : find the min of array from 0 to 4 which is 2, then swap it with the element 0.  $arr = [2, 10, 6, 5, 3]$  -  $i=1$ : find the min of array from 1 to 4 which is 3, then swap it with the element 1.  $arr = [2, 3, 6, 5, 10]$  -  $i=2$ : find the min of array from 2 to 4 which is 5, then swap it with the element 2.  $arr = [2, 3, 5, 6, 10]$  -  $i=3$ : find the min of array from 3 to 4 which is 6, then swap it with the element 3.  $arr = [2, 3, 5, 6, 10]$  -  $i=4$ : find the min of array from 4 to 4 which is 10, then swap it with the element 4.  $arr = [2, 3, 5, 6, 10]$

So, basically what you need to do is: 1. define a integer to separate the sorted and unsorted array 2. find the min number inside the unsorted array and put it into the sorted area 3. repeat step2 until all elements are sorted.

Please implement the sorting algorithm in `exer2_1.cpp` according to the above information, such that the program will output the array in ascending order.

For compiling and running the code, using the following commend lines:

```
# compile
g++ -o exer_run2_1 exer2_1.cpp
# run
./exer_run2_1
```

## Additional Exercise

**NO BONUS POINTS, just for fun :)**

### The Rod Cutting Problem description

Given a rod of length  $n$  and prices  $p_i$  for  $i = 1, \dots, n$ , where  $p_i$  is the price of a rod of length  $i$ . Find a way to cut the rod to maximize total revenue.

For example,  $n = 4, p_n = \{2, 5, 6, 11\}$ . For the length of 4 rod, there are the following cutting methods:

- 4 (11) - 3+1 (8) - 2+2 (10) - 2+1+1 (9) - 1+1+1+1 (8)

So, for rod length of 4 the max revenue will be 11

Another example,  $n = 4, p_n = \{3, 6, 7, 11\}$ . For the length of 4 rod, there are the following cutting methods:

- 4 (11) - 3+1 (10) - 2+2 (12) - 2+1+1 (12) - 1+1+1+1 (12)

So, for rod length of 4 the max revenue will be 12

## How to Solve the Problem

We will use **dynamic programming** to solve the problem.

Let  $r_i$  be the maximum revenue obtainable from cutting a rod of length  $i$ . Our goal is finding  $r_n$ .

For a rod of length  $n$ , assume the “first” cut length is  $j$ , which leaves a piece of length  $n - j$ . And the total value would be  $p_j + r_{n-j}$ . So, for  $j$  from 1 to  $n$ , the max value of  $p_j + r_{n-j}$  would be the max revenue.

Then, how to find  $r_i$  for all  $i$  in  $\{0, 1, \dots, n\}$ ? Since  $r_n$  can be driven from  $\{r_1, \dots, r_{n-1}\}$ ,  $r_{n-1}$  can also be driven from  $\{r_0, \dots, r_{n-2}\}$ . And we already know  $r_0 = 0, r_1 = p_1$ , then we can get  $r_2, \dots, r_n$ .

If you need a more detailed explanation, please read this document from COMP3711 (**pseudo code inside :o**).

## Your Mission

Now you know how to solve the problem, please implement the code in `exer2_2.cpp` such that the program will output the maximum revenue.

We have provided the skeleton code `exer2_2.cpp`, you only need to complete the missing part, input and output is already done. For compiling and running the code, using the following command lines:

```
# compile
g++ -o exer_run2_2 exer2_2.cpp
# run
./exer_run2_2
```

If you want to know more about compiling with `g++`, click [here](#)

**DO NOT FORGET TO COMMIT AND PUSH YOUR CODE :)**