****



Mini Project Report

On

**“Landscape Generator”**

Under the guidance of

**Mrs. Mrunal Deshpande**

**School of Computer Science & Engineering**

**Academic year: 2021 - 22**

**CERTIFICATE**

This is to certify that the mini project report entitled

**“Landscape Generator”**

submitted by

**Yash Sunil Desai (Roll No. 35 )**

**Harsh Uttam Patil (Roll No. 36)**

**Amol Duryodhan More (Roll No. 37)**

**Shivraj Mahesh Patil (Roll No. 40)**

In the partial fulfilment for the Semester IV of S. Y. B. Tech of Computer Science & Engineering is a record of work carried out by the students mentioned above under the guidance and supervision of Mrs. Mrunal Deshpande during the academic year 2021-22.

**Place: SGU, Atigre**

**Date:**

**Mrs. Mrunal Deshpande Dr. B. Suresh Kumar**

**Mini Project Guide Head-SOCSE**

**ACKNOWLEDGEMENT**

We express our sincere thanks to **Mrs. Mrunal Deshpande** and **Dr. B. Suresh Kumar, Head-SOCSE**whose supervision, inspiration and valuable guidance , helped us a lot to complete our mini project work.

Their guidance proved to be the most valuable to overcomeall the hurdles in the fulfilment of this mini project work.

Last but not the least, this acknowledgement would be incomplete without rendering our sincere gratitude to all those who have helped us in the completion of mini project work.

Sincerely,

1. **Yash Sunil Desai (Roll No. 35 )**
2. **Harsh Uttam Patil (Roll No. 36)**
3. **Amol Duryodhan More (Roll No. 37)**
4. **Shivraj Mahesh Patil (Roll No. 40)**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Chapter** | **Page No.** |
| 1 | Introduction | 01 |
| 2 | Problem Statement | 02 |
| 3 | Literature Review | 03 |
| 4 | Objectives | 04 |
| 5 | Methodology |  |
|  | 5.1 General Method | 05 |
|  | 5.2 Algorithm | 06 |
|  | 5.3 Flowchart | 07 |
| 6 | System Requirements | 08 |
| 7 | Results | 09 - 11 |
| 8 | Conclusion | 12 |
| 9 | References | 13 |

**CHAPTER 1**

**Introduction**

We are working on a computer game for generating a realistic landscape.

Our algorithm inspires by geological processes.

Firstly, we have started our algorithm by generating a simple flat landscape.

It modifies ups and downs of landscapes repeatedly.

The algorithm starts with a flat landscape and repeatedly modifies it by lifting or lowering continuous blocks, thus forming lifted blocks and lowered blocks.

The blocks to be lifted or lowered are selected at random.

Our task is to interpret any sequence of such modifications and output the resulting landscape.

**CHAPTER 2**

**Problem Statement**

Interactive Creative Players Collective (ICPC) is working on a new computer game for which they want to generate realistic landscapes. One of the ICPC engineers proposed an algorithm inspired by geological processes.

The algorithm starts with a flat landscape and repeatedly modififies it by lifting or lowering continuous blocks, thus forming *horsts* (lifted blocks) and *grabens* (lowered blocks). The blocks to be lifted or lowered are selected at random. ICPC hopes to obtain realistic landscapes this way.

Our task is to interpret any sequence of such modififications and output the resulting landscape. The landscape is represented by a sequence of *n* integer height values, one for each integer point from 1 to *n* on the *x*-axis.

**CHAPTER 3**

**Literature Review**

* C++ Basic Concepts

<https://www.javatpoint.com/cpp-tutorial>

* C++ Control Statement

<https://www.javatpoint.com/cpp-tutorial>

## C++ IF-else Statement:

The C++ if-else statement also tests the condition.

It executes if block if condition is true otherwise else block is executed.

# C++ For Loop:

The C++ for loop is used to iterate a part of the program several times.

# C++ While loop:

In C++, while loop is used to iterate a part of the program several times.

If the number of iteration is not fixed, it is recommended to use while loop than forloop.

# C++ Vector

A vector is a sequence container class that implements dynamic array, means size automatically changes when appending elements. A vector stores the elements in contiguous memory locations and allocates the memory as needed at run time.Syntax

vector<object\_type> v1;

# C++ | Nested Ternary Operator:

<https://www.geeksforgeeks.org/c-nested-ternary-operator/>

**[Ternary operator](https://www.geeksforgeeks.org/cc-ternary-operator-some-interesting-observations/)**also known as conditional operator uses three operands to perform operation. This ternary operator is similar to if-else statement.  
**Syntax :**   
 **op1 ? op2 : op3;**

**Nested Ternary operator:** Ternary operator can be nested. A nested ternary operator can have many forms like : a ? b : c or a ? b: c ? d : e ? f : g ? h : i

* Landscape Generator

<https://www.youtube.com/watch?v=B6yNhtNehE0>

**CHAPTER 4**

**Objectives**

The main Objective of Landscape generator is-

1. To generate a realistic landscape for a computer Game.
2. To interpret any sequence of such modifications like the lifting or lowering landscape outputs.
3. The main goal is to calculate the result height of every position of on our line.

**CHAPTER 5**

**Methodology**

**5.1 General Method**

Initially the height is 0 at all *n* points. This flat shape is subjected to a sequence of modififications. Each modifification applies one of the following four operations with two integer parameters *x*1 *≤ x*2:

R: Raise – increase the height by 1 at all points between *x*1 and *x*2 inclusive.

D: Depress – decrease the height by 1 at all points between *x*1 and *x*2 inclusive.

H: Hill – add a new linearly shaped hill between *x*1 and *x*2.

V: Valley – add a new linearly shaped valley between *x*1 and *x*2.

Adding a hill to the current landscape works as follows. The heights at points *x*1 and *x*2 are increased by 1. If *x*2 *− x*1 *>*1, the heights at points *x*1 + 1 and *x*2 *−* 1 are increased by 2. If *x*2 *− x*1 *>*3, the heights at points *x*1 + 2 and *x*2 *−* 2 are increased by 3, and so on. Adding a valley works in the same way except the heights are decreased instead. The maximal change of height happens in the middle between *x*1 and *x*2. If *x*2 *− x*1 is odd, there will be two neighboring points with maximal change, otherwise just one.

**5.2 Algorithm**

Step 1: Create a main function.

Step 2: Take number of points as input.

Step 3: Take number of modifications as input(int K)

Step 4: For K modifications

1) input character C

2) input operation X1 and X2

Step 5: if operation (op) is equals to R then ,increase the height by 1

Else do Step 6

Step 6: if else if operation (op) is equals to H then ,decrease the height by 1

Else do Step 7

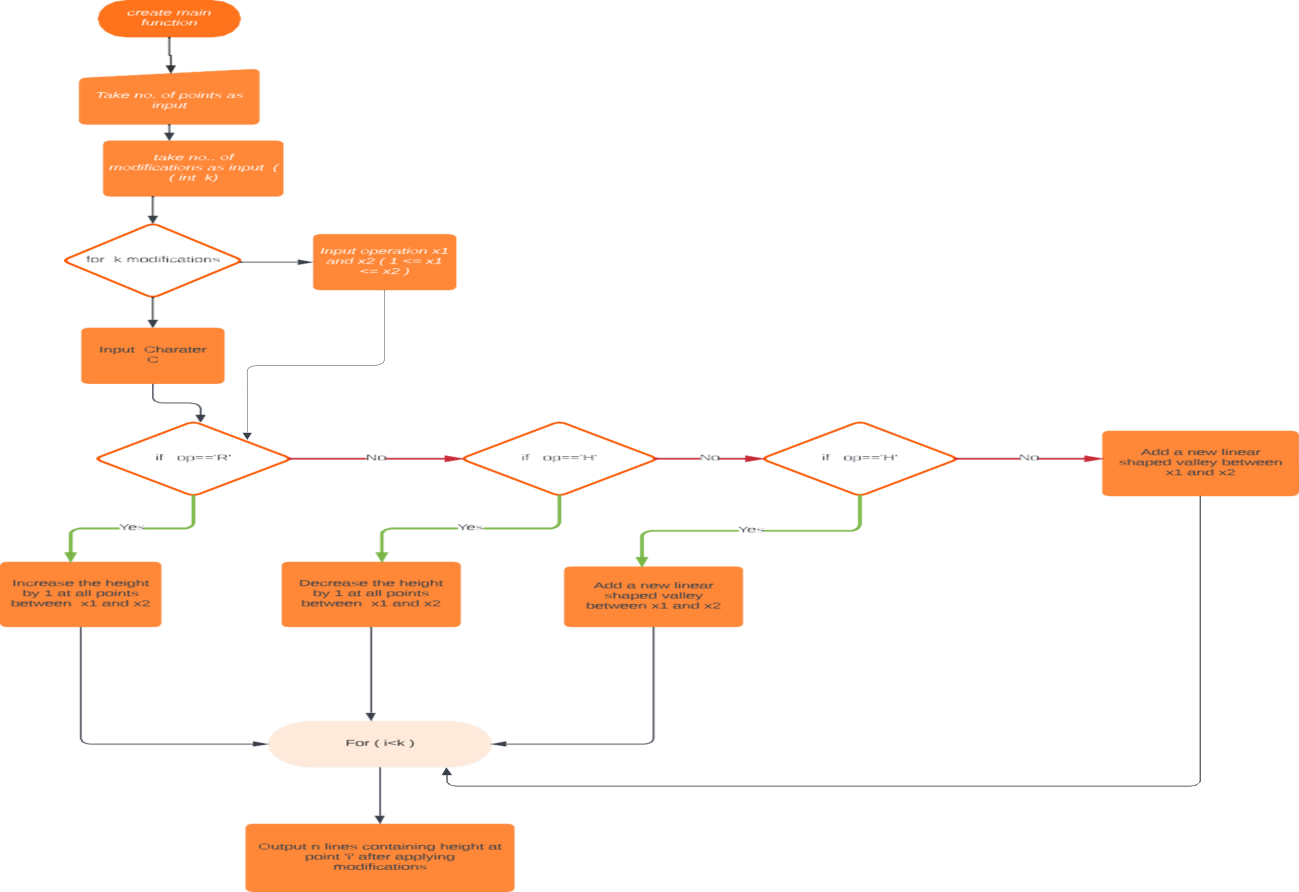
Step 7: if operation (op) is equals to V then ,linear shaped valley between X1 and X2

Step 8: For number of iterations less than number of modifications

Step 9: Outputing lines containing height at point of all iterations after applying all modifications

Step 10: Stop

**5.3 Flowchart**



**CHAPTER 6**

**System Requirements**

* **Software requirements:**
* C++ compiler
* **Hardware**

**requirement:**

* Processor: Inteli3/i5/i7
* RAM minimum 4 GB
* Hard disk minimum 15 GB
* System type 32bit Operating System
* Internet connection

**CHAPTER 7**

**Results**

**Inputs:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

int N, K;

while (cin >> N >> K)

{

vector<int> xd(N+10);

for (int i = 0; i < K; i++)

{

char ch;

int X1, X2;

cin >> ch >> X1 >> X2;

if (ch == 'R' || ch == 'D') {

xd[X1-1] += (ch == 'R' ? 1 : -1);

xd[X1] -= (ch == 'R' ? 1 : -1);

xd[X2] -= (ch == 'R' ? 1 : -1);

xd[X2+1] += (ch == 'R' ? 1 : -1);

}

else

{

xd[X1-1] += (ch == 'H' ? 1 : -1);

xd[(X2+X1 )/2] -= (ch == 'H' ? 1 : -1);

xd[(X2+X1+1)/2] -= (ch == 'H' ? 1 : -1);

xd[X2+1] += (ch == 'H' ? 1 : -1);

}

}

int x = 0, d = 0;

for (int i = 0; i < xd.size(); i++) {

x += d;

d += xd[i];

if (i >= 1 && i <= N) cout << x << endl;

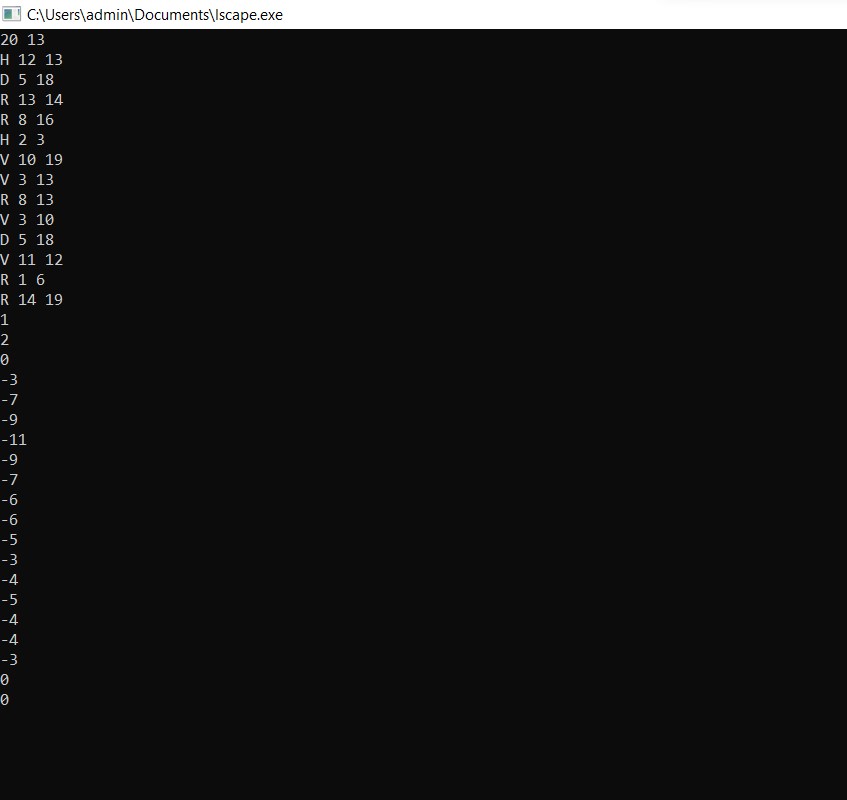
}

}

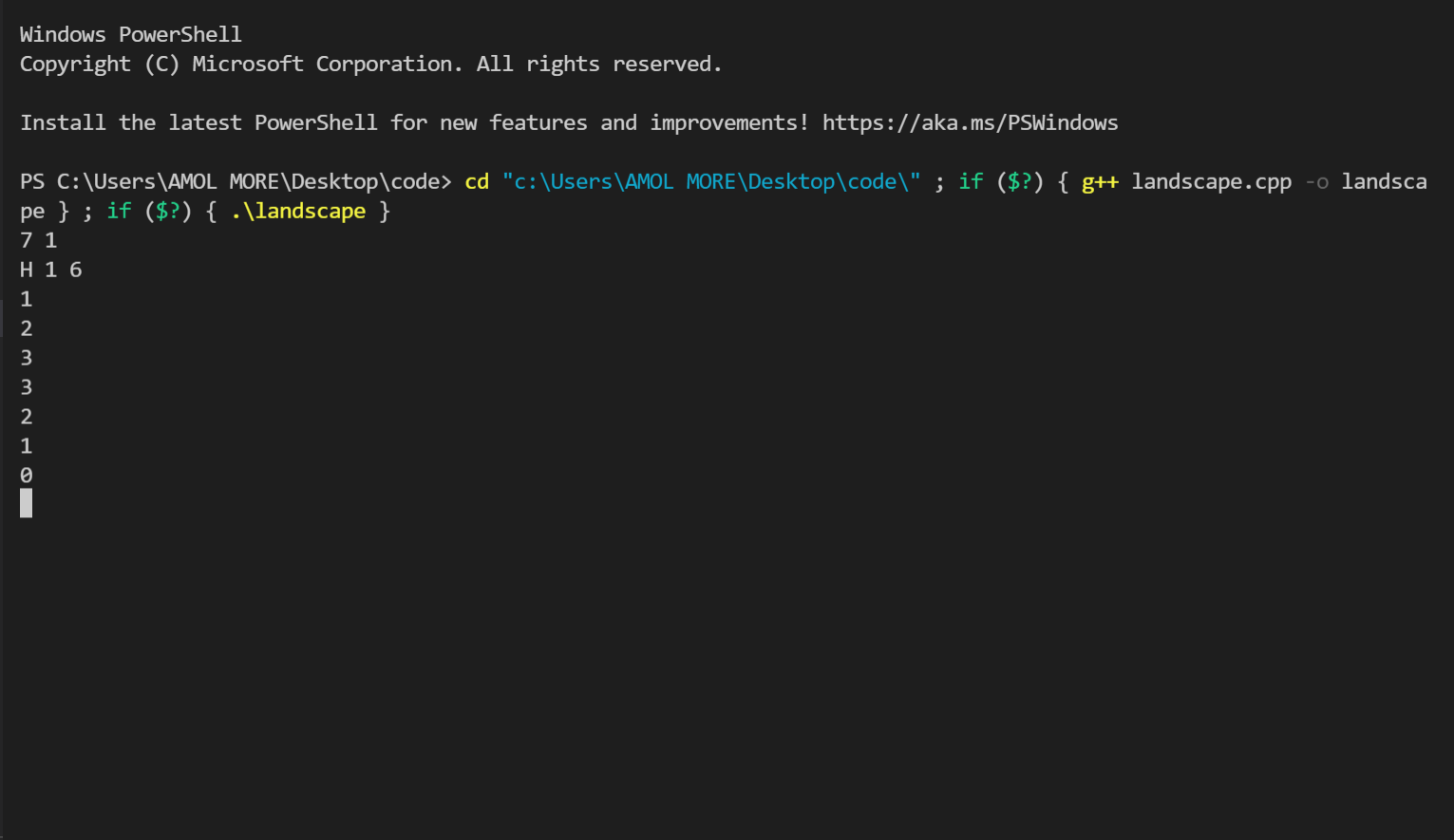
}

**Outputs:**

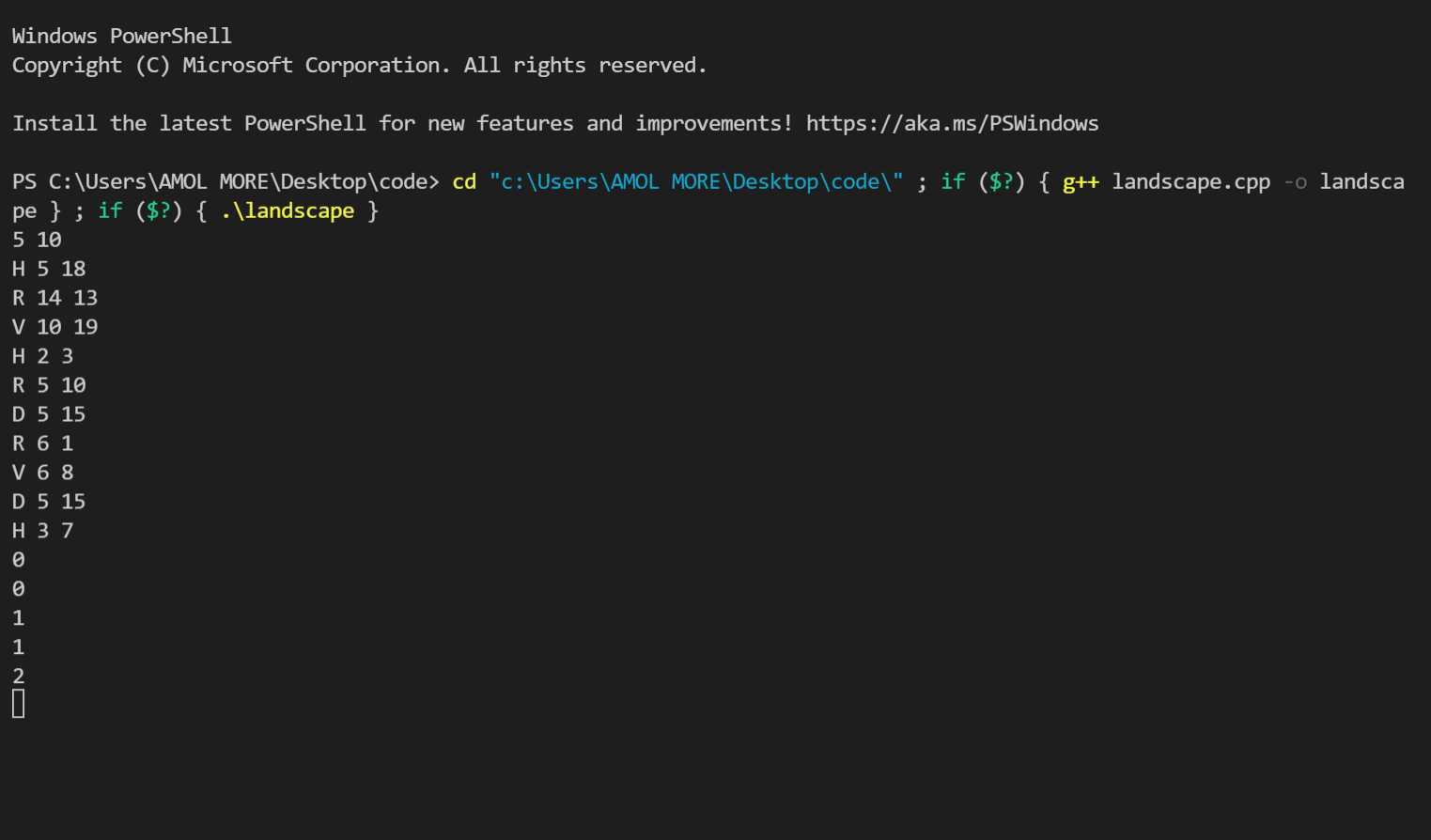
**1.**

****

**2.**

****

**3.**

****

**CHAPTER 8**

**Conclusion**

We have successfully implemented coding for Landscape Generator that will interpret any sequence of such modififications and output the resulting landscape.Thus we have successfully found the logic behind the problem statement.

**CHAPTER 9**

**REFERENCES**

1. [https://icpc.global/worldfinals/problems/2020+ACM](https://icpc.global/worldfinals/problems/2020+ACM-ICPC+World+Finals/icpc2020.pdf)-[ICPC+World+Finals/icpc2020.pdf](https://icpc.global/worldfinals/problems/2020+ACM-ICPC+World+Finals/icpc2020.pdf)

[2]<https://www.youtube.com/watch?v=B6yNhtNehE0>

[3] <https://www.javatpoint.com/cpp-tutorial>

[4]<https://www.geeksforgeeks.org/c-nested-ternary-operator/>