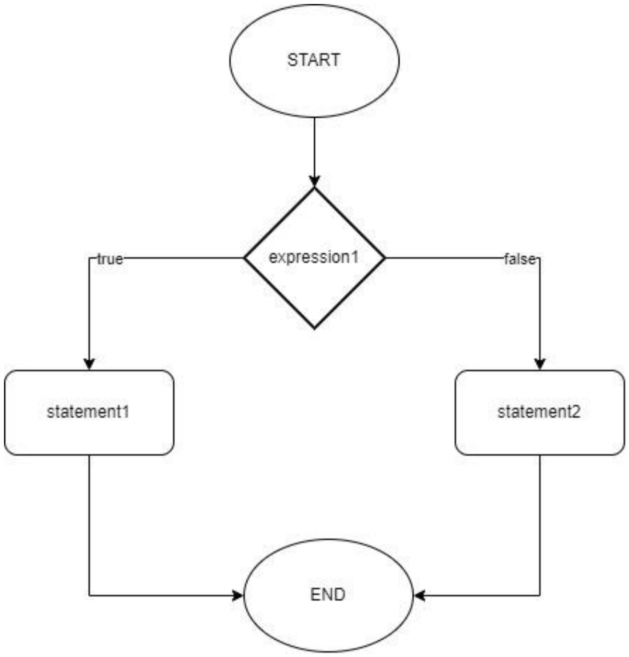
Data Structures – Spring 2023
Project 5
Due: April 19th 2023 at 11:59 PM

Objective:
The goal of this project is to write a program that generates code for a given flowchart.

Description:
Code generation from a flowchart is the process of automatically translating a flowchart diagram into executable code in a programming language. It is a technique commonly used in software development, especially in the development of graphical programming languages or visual programming environments.

The code generation process starts with the design of a flowchart that describes the algorithm or process to be implemented. The flowchart typically consists of various symbols such as start and end nodes, decision/if nodes, and process blocks, each representing a specific action or condition. The flowchart can be represented in many ways but here we represent it as an adjacency list and that is given as an input. This adjacency list is stored as a vector of vectors.

Once the flowchart is designed, the code generation process involves analyzing the flowchart and automatically generating code that implements the same algorithm. The flowchart nodes can either be START, END, IF or BLOCK. The information about the flowchart is given as an input explained below. All the information about the each node is stored as an array of node objects. Class node has fields to store information about a single node. The following is the flowchart represented in input 1. There can be more nodes and blocks given including nested if conditions.



Input file:

```
5

0 START
1 IF expression1
2 BLOCK statement1
3 BLOCK statement2
4 END

0 1
1 2 3
2 4
3 4
4
```

The first number represents the number of nodes in the flowchart. Later for each of the nodes, we the input file gives the type, and the expression or statement depending on the type of the node. This is followed by the adjacency list. Below is an explanation:

- 0 is the START node and its out arrow goes to node 1.
- 1 is the IF condition for which the true goes to node 2 and false flow goes to node 3. Note that always the first node will represent the true flow and the second node will represent the false flow.
- 2 is a BLOCK that has an statement1 that flows to the END node 4.
- 3 is a BLOCK that has an statement2 that flows to the END node 4.
- 4 is the END node itself so there is no flow going out.

A small sample of how the output generated code should look like is given in the boilerplate code including the class and main function. Formal output for the autograder will be released soon. Also note that the convertFlowChart() is a function and is not a part of any class. You will have to use a stack here to keep track of what flow you are currently at to be able to print the close and open parenthesis properly. You are not allowed to use a recursive approach.

Redirected input:

Redirected input provides you a way to send a file to the standard input of a program without typing it using the keyboard. To use redirected input in Visual Studio environment (on windows), follow these steps: After you have opened or created a new project, on the menu go to project, project properties, expand configuration properties until you see Debugging, on the right you will see a set of options, and in the command arguments type <"input filename". The < sign is for redirected input and the input filename is the name of the input file (including the path if not in the working directory).

If you use macOS or linux (or windows using powershell), you may use the terminal to compile and run your program using g++. This is the compiler that we use in the autograder on GradeScope (you may need to install g++ on your computer). Once you installed g++, you may compile your program using the following command:

```
g++ project5.cpp -o p5
```

You may then run the program with the redirected input using the following command:

./p5 < input1.txt

For powershell,

Get-Content input1.txt | ./p5

Make sure your program and your input file are in the same folder.

You may also use Google CoLab as an alternative and a sample file is given along with the project. This is an ipynb file which has to be opened in Google CoLab.

A simple program that reads the input file (one number) and displays everything that was read is given below.

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

int main()
{
  int num = 0;
  cout << "Hello World" << endl;
  cin >> num;
  cout << num;
  return 0;
}
```

Sample output file for corresponding input files will be released. The input1.txt file given is a simple input file that will help you understand the project, more complicated ones will be released later and used for grading.

Submission:

Submission will be through GradeScope. Your program will be autograded with test cases and then hand graded to check for documentation and if you have followed the specifications given. You may submit how many ever times to check if your program passes the test cases provided. Test cases will be released at the beginning and later other test cases will be released while grading. For the autograder to work, the program you upload must be named as **project5.cpp**. This will be the only file you will submit. Make sure it is well commented.

Constraints:

1. In this project, the only header you will use the header files that are given to you in the boiler plate code. Using other or excess header files will be subject to a heavy grade penalty for the project.
2. None of the projects is a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.

- Please review academic integrity policies in the modules.

How to ask for help:

1. Check FAQ discussion board of the project first.
2. Email the TAs with your precise questions.
3. Upload your well commented and clean program to CodePost.
    a. This is a website which is used to share code with your TA only.
    b. You will upload your program and I can view it and comment on it.
    c. Here is the invite link to our class for the summer.
    d. Once you join the class on CodePost, you can upload your program to the Project 1 assignment.

        Note: Your program will **not** be auto graded at CodePost, this is just for when you want to ask a question and a place where I can look at your program and comment on it. CodePost is great for live feedback. GradeScope is the place where you should submit and where your program will be autograded.