Soni Dilkhush S

BVM Engineering College

Vidhyanagar

dilkhush57@gmail.com

**Farematrix’s Program Documentation:**

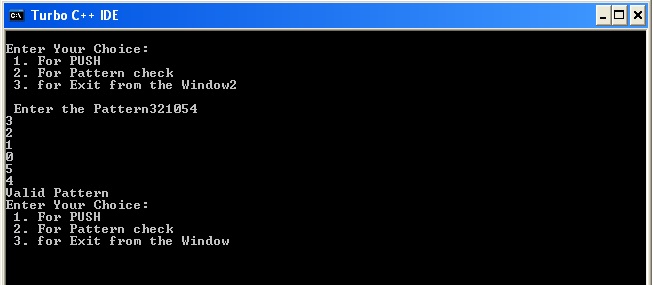
**Program1 :** Program 1 : Write a program to simulate the push/pop functionality of a Stack. The requirement are  
  
a) It should be possible to PUSH numbers (0 - 9) on the stack only in the proper sequence. Meaning, the allowed sequence is PUSH 0, PUSH 1, PUSH 2, PUSH 3, PUSH 4, PUSH 5, PUSH6, PUSH 7, PUSH 8, PUSH 9. It should not be possible to skip numbers in between. PUSH 2, PUSH 4 is not allowed.  
  
b) Given a random pattern of numbers, the program should be able to validate if it is possible to generate the given pattern using a number of PUSH - POP operations. For example: the pattern 4 3 2 1 0 9 8 7 6 is achievable using the following PUSH - POP sequence.

**Solution:** This program is implemented in C. In this program I use simply push pop functionality by simply creating two functions called Push and Pop.

Push will push the element on the stack and pop will pop the element from the stack but it check whether stack is empty or not.

For check the **part a** of the program simply give the choice 1 on command prompt. Then it will ask u to enter the number. The new number that is to be push is store in variable **n** and the previously entered number is store in **s.** first time it simply push the element second time it check whether n = s+1 if condition is true them it allow u to Push the element otherwise generate the error that “Invalid PUSH”

**Part b** Enter the choice 2 for check the given pattern is valid or not. At the first time when u enter the pattern it will go in the first if condition ( if(i==0 ) and push the element till the first character of the given pattern not come and pop that element last element that is to be pop is stored in the **temp** variable. Next time for the increment value of I the program will check following condition if(temp == s[i]-'0') or if(temp < s[i]-'0') according to that push and pop will done. Every time the poped element will be display on command prompt that is seen in the screen short.



Finally if the temp ==1 then it will display “Valid Pattern” otherwise “Invalid Pattern”.

For exit enter the choice 3.

**Program 2**: Game of Elimination.

The rules of the game are:  
  
a) Suppose there are M kids. They stand in a circle and pick a random number N which is less than M.  
  
b) They start counting from 1 to N and the kid at position N is eliminated. They resume counting from position N + 1 starting from 1.  
  
c) This continues until there is only one kid left. The last Kid standing is declared the Winner.

**Solution:**

This program also written in C using **Circular Linklist**.

Here player is a structure variable. **head** represent the starting pointer of the list and **current** represent where the pointer is currently.

void freelist() function free the node. getnode() is useful for creating a new node dynamically.

void list\_creation(int n) here **n**  is the no of player that has been enter on command prompt and new node is add after the previously generated node.

if n<=1" prompt generate error that “There Should be at least 2 Players for this game"

void display() display the player number. and tail function is use for finding the last element.

The remaining player after each elimination is is find and display by “remaining\_after\_died(int by\_n)” where **by\_n** is the key by which the players eliminate. for (j=1;j<by\_n;j++) this loop find the by\_n th player from the list. When this player found just remove it from the list and adjust the pointer.

Here enter choice

**1**. For player list creation

**2**. For Display the players list

**3**. For Remove player

**4**. For Exit from Game.