**UCS 2312 Data Structures Lab**

**Assignment 4: StackADT and its application**

Create an ADT for the stack data structure with the following functions. stack*ADT* will have the

integer array, top and size. [CO1, K3]

1. createStack(top) – initialize size and top with -1
2. push(top,data) – push data into the stack if stack is not full. Print a message when stack

is

full

1. pop(top) – decrements the top by 1
2. peek(top)– returns the element at top, if stack is not empty, otherwise returns -1
3. isEmpty(top) – returns 1 if stack empty, otherwise returns 0
4. isFull(top) – returns 1 if stack full, otherwise returns 0

Test the operations of stackADT with the following test cases

|  |  |
| --- | --- |
| **Operation** | **Expected Output** |
| peek(top) | Empty |
| push(top,2) | 2 |
| push(top,4) | 4, 2 |
| push(top,6) | 6, 4, 2 |
| push(top,8) | Full |
| pop(top) |  |
| peek(top) | 4 |
| peek(top) | 4 |
| pop(top) |  |
| pop(top) |  |
| peek(top) | Empty |
| pop(top) |  |
| pop(top) |  |
| push(top,11) | 11 |
| peek(top) | 11 |

Best practices to be followed:

* Design before coding
* Usage of algorithm notation
* Use of multi-file C program
* Versioning of code

**Application using Stack**

1. Apply the stack to check whether the given input is balanced or not?

Examples:

[[ ), [ ()() ], ( ( [][] ) )

1. Convert the given decimal number into binary using stack Example: 14

Ans: 1110

1. Understand the following procedure to use Stack and trace it for the input: input: [34, 3, 31, 98, 92, 23] and write an algorithm.
   * Create a temporary stack say **tmpStack**.
   * While input stack is NOT empty do this:
     + Pop an element from input stack call it **temp**
     + while temporary stack is NOT empty and top of temporary stack is greater than temp,

pop from temporary stack and push it to the input stack

* + - push **temp** in temporary stack • Print the numbers in tmpStack Predict the output ??? = [ ]

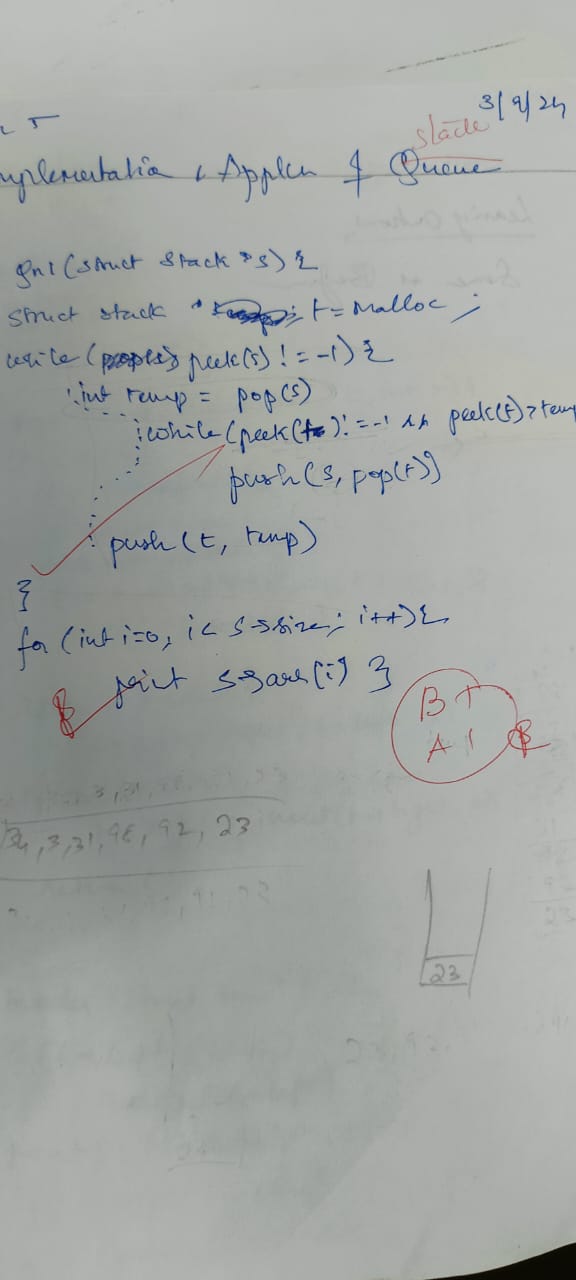
1. Given an array of integers temperatures represents the daily temperatures, return *an array* answer *such that* answer[i] *is the number of days you have to wait after the* ith *day to get a warmer temperature*. If there is no future day for which this is possible, keep answer[i] == 0 instead. (use stack)

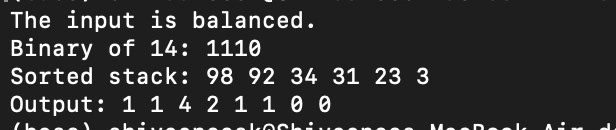
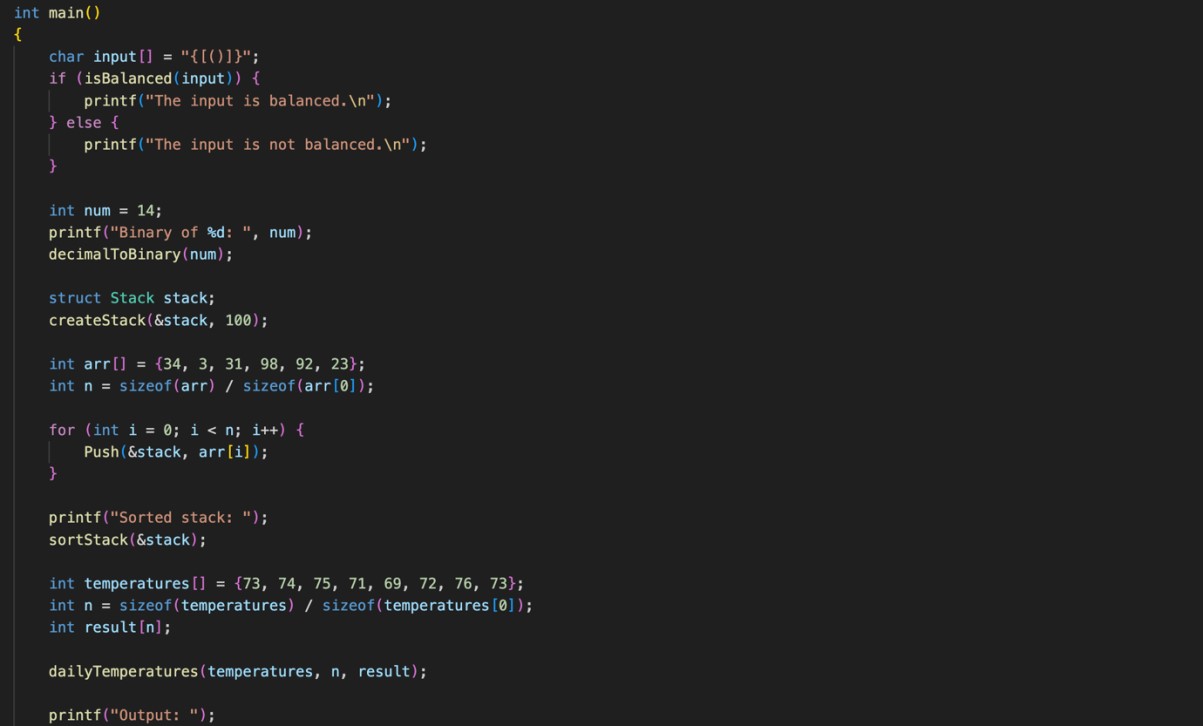
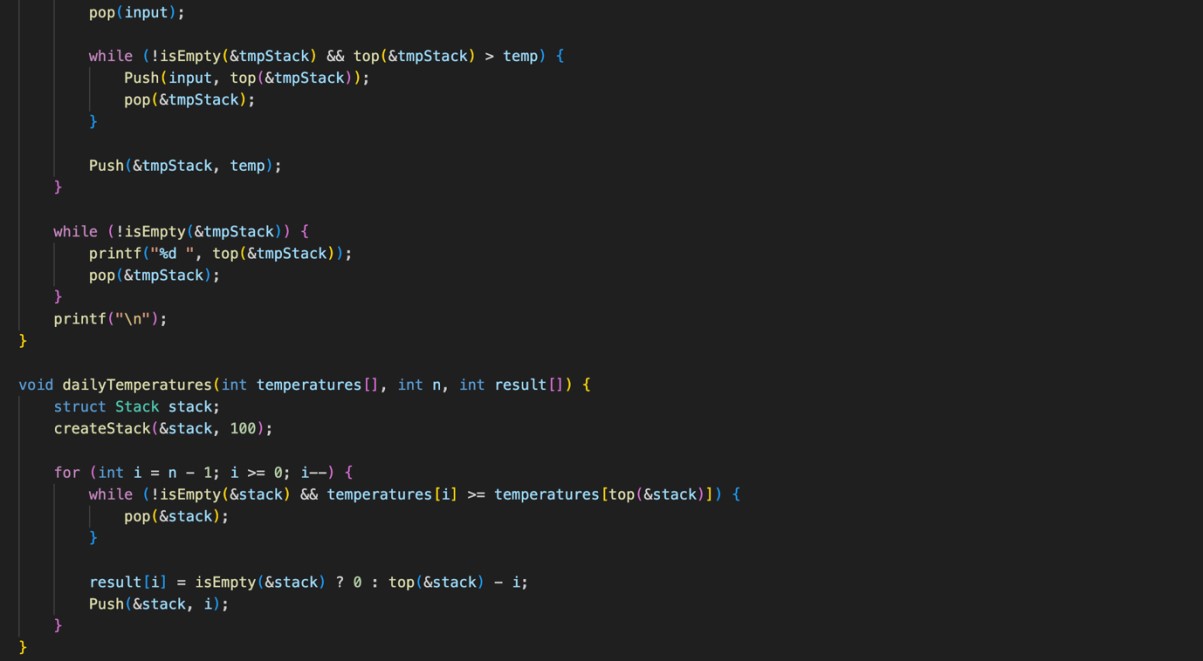
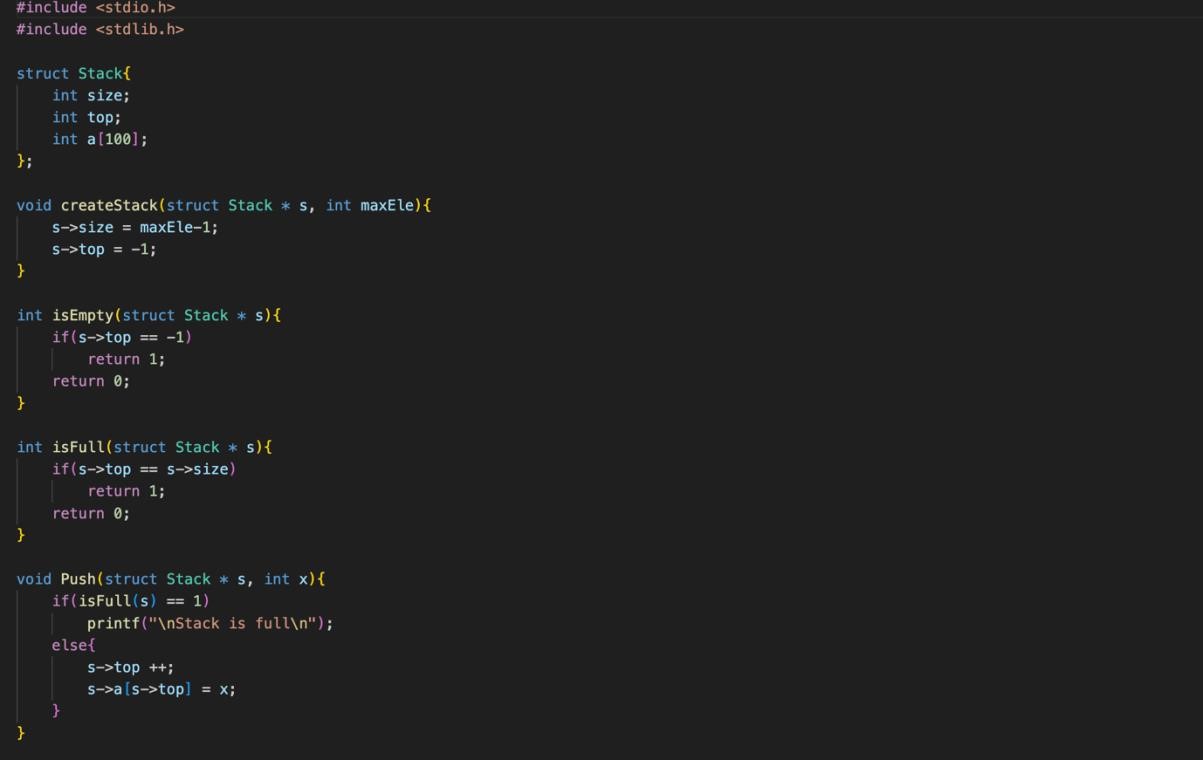
**Example 1:**

**Input:** temperatures = [73,74,75,71,69,72,76,73] **Output:** [1,1,4,2,1,1,0,0] **Example 2:**

**Input:** temperatures = [30,40,50,60]

**Output:** [1,1,1,0]





Technical Outcomes

|  |  |  |
| --- | --- | --- |
| Design | 2 | Needs improvement |
| Understanding of DS | 2 | Needs improvement |
| Use of DS | 3 |  |
| Debugging | 3 |  |

Best Practices

|  |  |  |
| --- | --- | --- |
| Design before coding | 2 | Needs improvement |
| Usage of Algo | 3 |  |
| Multifile | 3 |  |
| Versioning | 3 |  |