Experiment No.1

DATE / /

- Aim:- Implementation of stack using array for real world application.

- Objective

→ 1] To introduce the concepts of data structures and analysis procedure
2] To conceptualize linear data structures and its implementation for various real world application.

- Theory

→ 1] Introduction to data structure:
Data structure (DS) are a way of organizing and storing data in a computer such that it can be retrieved and used effectively. C data structure considers not only the elements stored but also their relationship to each other.

- Classification of Data structure

(1) Primitive data structure
(2) Non-Primitive data structure
- Non-Primitive data structure classify as:
- (1) Linear list and non linear list
Examples of non-linear list are Array, stack, Queue, lists.

(2) Introduction to stack
- A list means a linear collection of element, For eg:- Array
- A linear list that follows insertion and deletion at one end only is called stack.
- Elements in stack have the same data type and are ordered by when they were added.
- The only accessible element : TOP
- Also known as last in first out (LIFO) as the elements are removed in the opposite order of which they were added.

(3) Various operations (PUSH, POP, PEEP, CHANGE, DISPLAY, etc.)

- PUSH

Push operation refers to inserting an element in the stack. Since there's only one position at which the new element can be inserted the new element is inserted at the top of stack.

- POP

POP operation refers to remove an element from the of the stack (newest element in stack). The element is removed to the stack contains and the size of the stack is ~~removed~~ decreased by 1.

- PEEP

PEEP operation refers to a stack function that returns the value of the top most element of the stack without deleting that element from stack.

- CHANGE

Change operation in stack function that user can change or update the content of the specific element.

- DISPLAY

The display() function displays all the elements in the stack. It uses a for loop to do so. If there is no element in the stack, then stack is empty is printed.

④ Algorithm

- PUSH

Procedure PUSH (S, TOP, x). This procedure inserts an element x to the top of the stack which is represented by a vector S containing N elements with a pointer TOP denoting the top element in the stack

1. [Check for stack overflow]
   IF TOP > N
   then Write ("STACK OVERFLOW")
   Return
2. [Increment Top]
   TOP ← TOP + 1
3. [Insert element]
   S[TOP] ← x
4. [Finished]
   Return.

- POP

Function POP (S, TOP). This function removes the top element from a stack which is represented by vector S and returns this element. Top is a pointer to the top element of the stack.

1. [Check for underflow on Stack]
   If TOP = 0
   then Write ('Stack Underflow on POP')
   take action in response to underflow
   Exit
2. [Decrement Pointer]
   Top ← Top − 1
3. [Return former top element of stack]
   Return (S[TOP + 1])

- PEEP

Function PEEP (S, TOP, I) gives a vector S consisting of N element representing a sequentially allocated stack and a pointer TOP denoting the top element of the stack. The element is not deleted by this function

1. [Check for stack underflow]
   If TOP - I + 1 ≤ 0
   then Write ('Stack Underflow on Peep')
   take action in response to underflow
   Exit
2. [Return element from top of stack]
   Return (S[TOP - I + 1])

- CHANGE

Procedure change (S, TOP, X, I). As before a vector S (consisting of N elements) represents a sequentially allocated stack and a pointer TOP denotes the top element of the stack. This procedure changes the value of the I^{th} element from the top of the stack to value contained by X.

1. [Check for stack underflow]
   If TOP - 1 + I ≤ 0
   then Write ('Stack underflow on change')
   Return
2. [Change I^{th} element from top of Stack]
   S[TOP - I + 1] ← X
3. [Finished]
   Return

- Display

1. Check whether stack is EMPTY [TOP == -1]
2. If it is empty, then display "stack is empty" and terminate the function.
3. If it is NOT EMPTY, then define a variable 'i' and initialize with top. Display stack [i] value and decrement i value by one (i--)
4. Repeat above step until i value becomes '0'.

(5) Example :-

1] A real life example is a stack of plates. You can only take a plate from the top of the stack and you can only add a plate to the top of the stack. this explains that stack uses LIFO principle.

2] Deck of cards - We can place or remove from top of card.

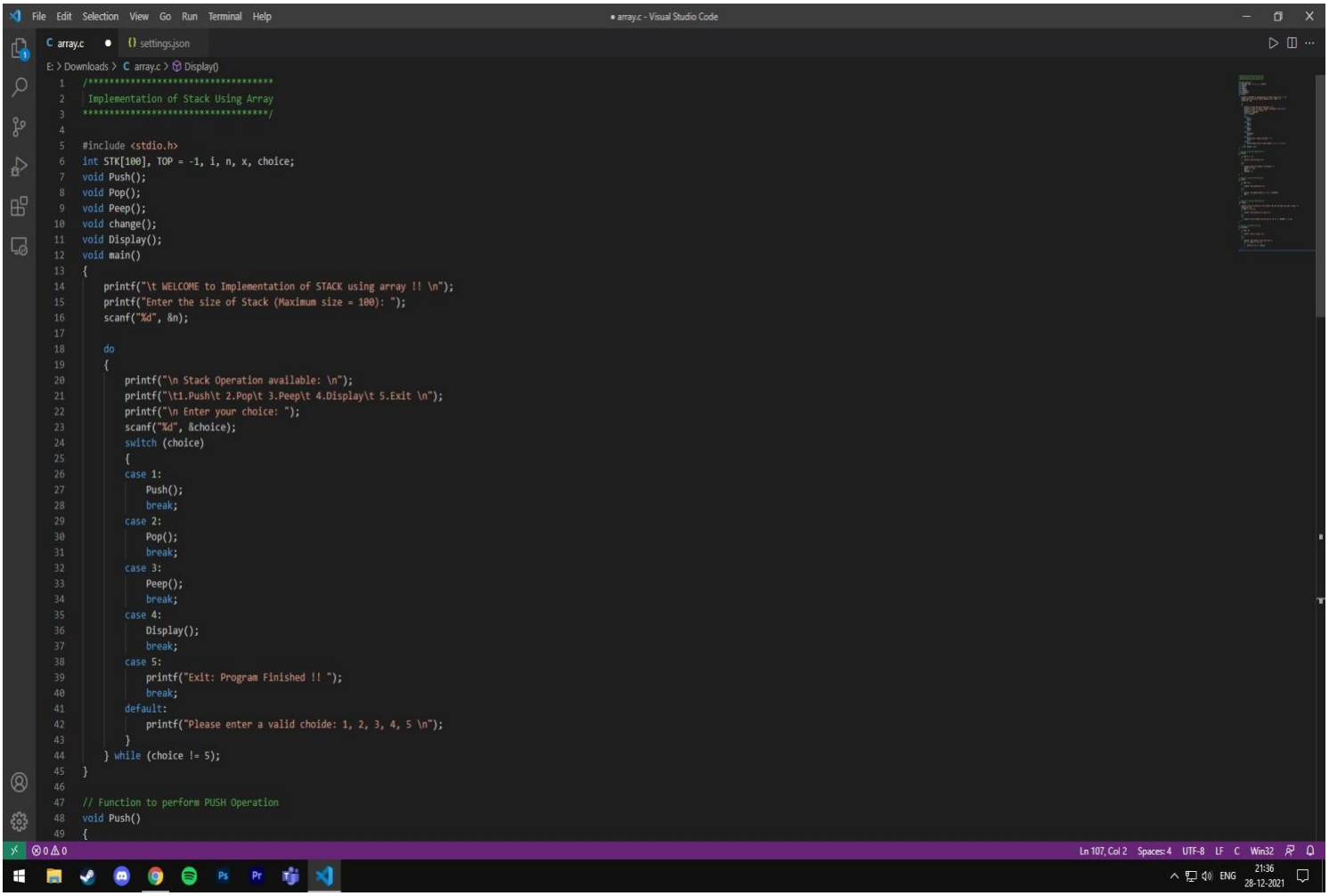
3] Backtracking - Process when you need to access the most recent data element in a series of element.

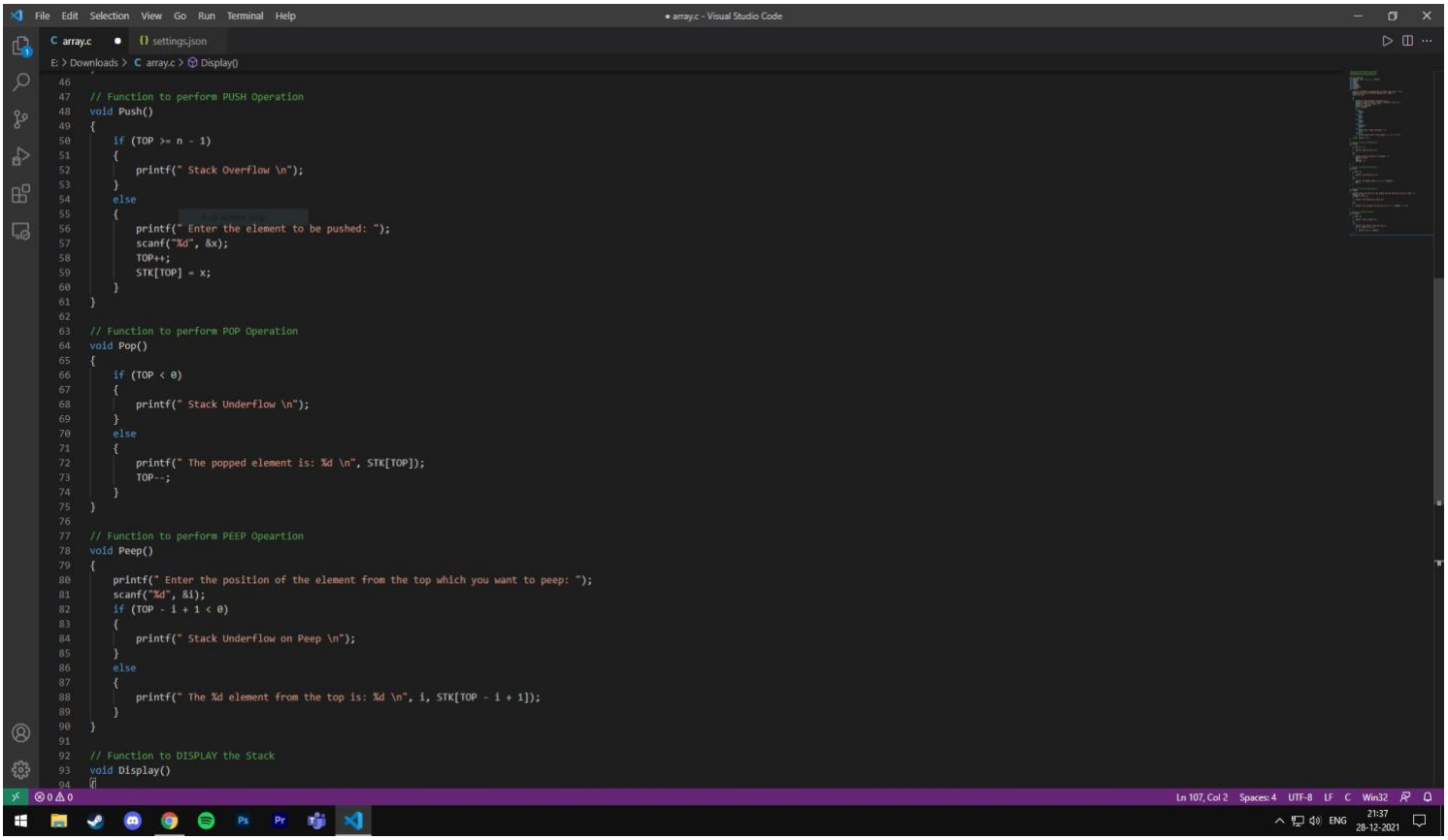
Conclusion:

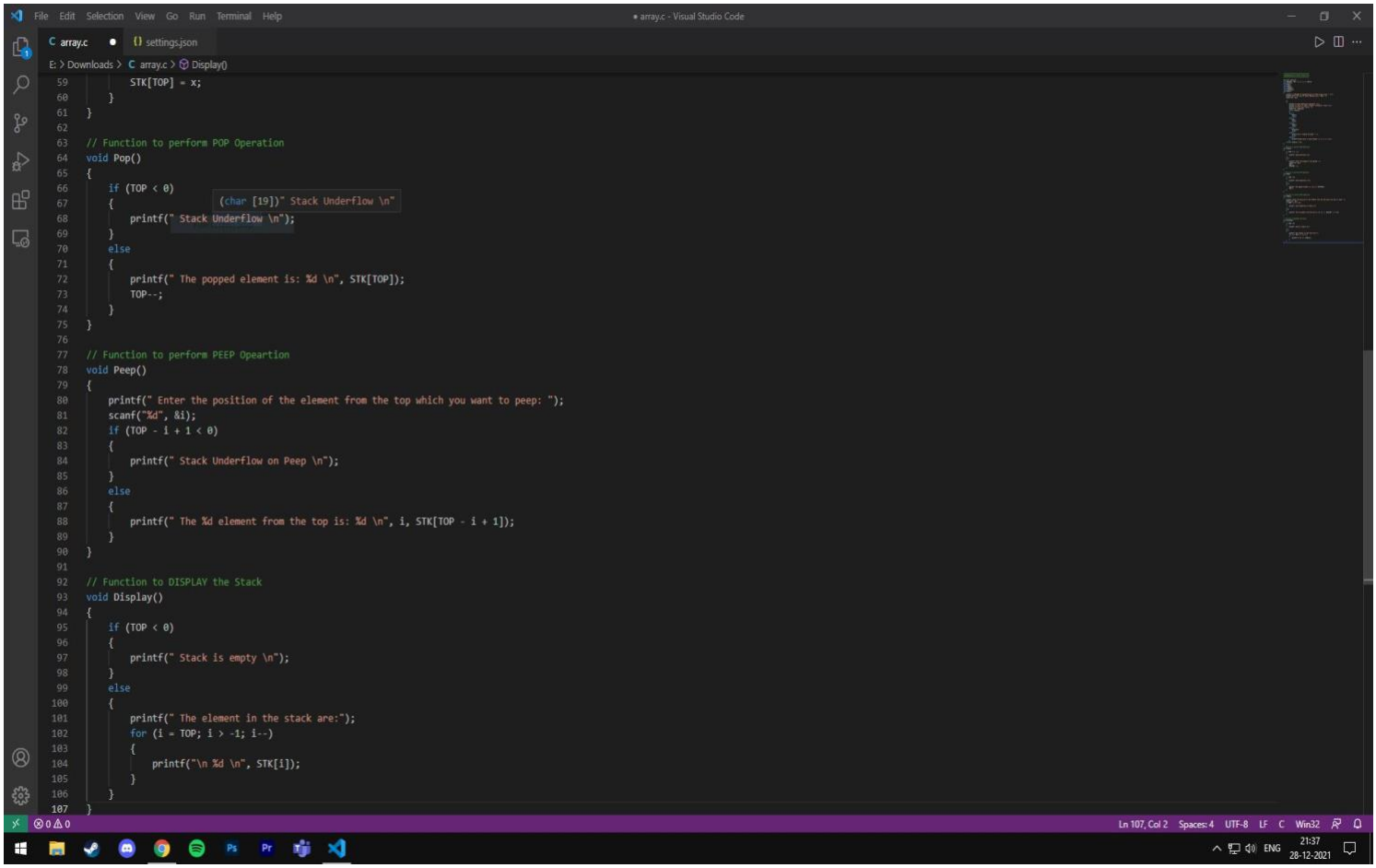
From this experiment we learnt that how to implement stack and the real life applications of stack and how to perform Push, Pop, peep, change, display operations in the stack with the help of algorithms

Output :-

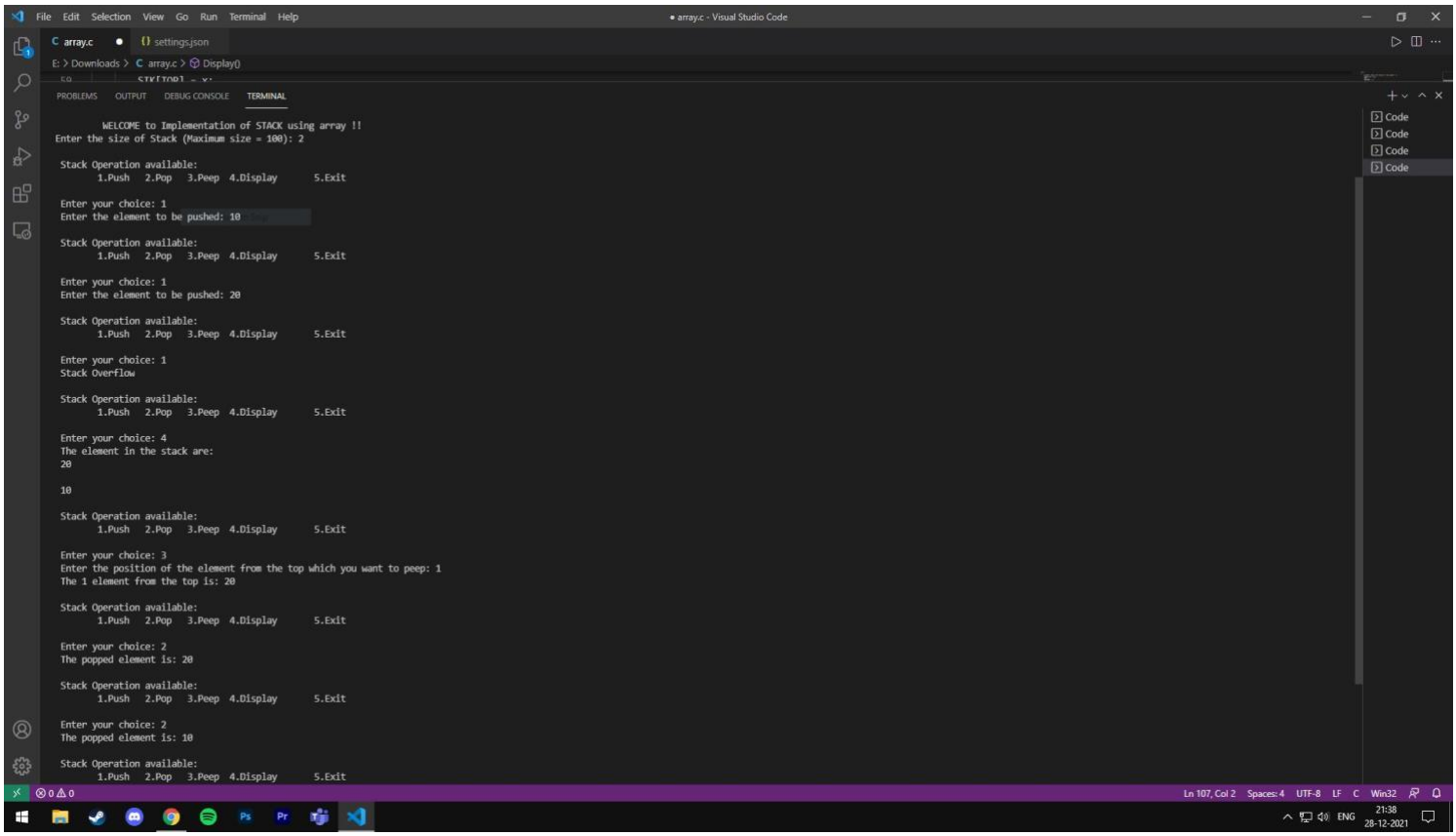
Apply the concepts of stack for real world application.


File Edit Selection View Go Run Terminal Help
• array.c - Visual Studio Code
array.c
settings.json
E: > Downloads > array.c > Display()
/**********************************
Implementation of Stack Using Array
**********************************/
#include <stdio.h>
int STK[100], TOP = -1, i, n, x, choice;
void Push();
void Pop();
void Peep();
void change();
void Display();
void main()
{
printf("\t WELCOME to Implementation of STACK using array !! \n");
printf("Enter the size of Stack (Maximum size = 100): ");
scanf("%d", &n);
do
{
printf("\n Stack Operation available: \n");
printf("\t1.Push\t 2.Pop\t 3.Peep\t 4.Display\t 5.Exit \n");
printf("\n Enter your choice: ");
scanf("%d", &choice);
switch (choice)
{
case 1:
Push();
break;
case 2:
Pop();
break;
case 3:
Peep();
break;
case 4:
Display();
break;
case 5:
printf("Exit: Program Finished !! ");
break;
default:
printf("Please enter a valid choide: 1, 2, 3, 4, 5 \n");
}
} while (choice != 5);
}
// Function to perform PUSH Operation
void Push()
{
Ln 107, Col 2 Spaces: 4 UTF-8 LF C Win32
ENG 21:36 28-12-2021

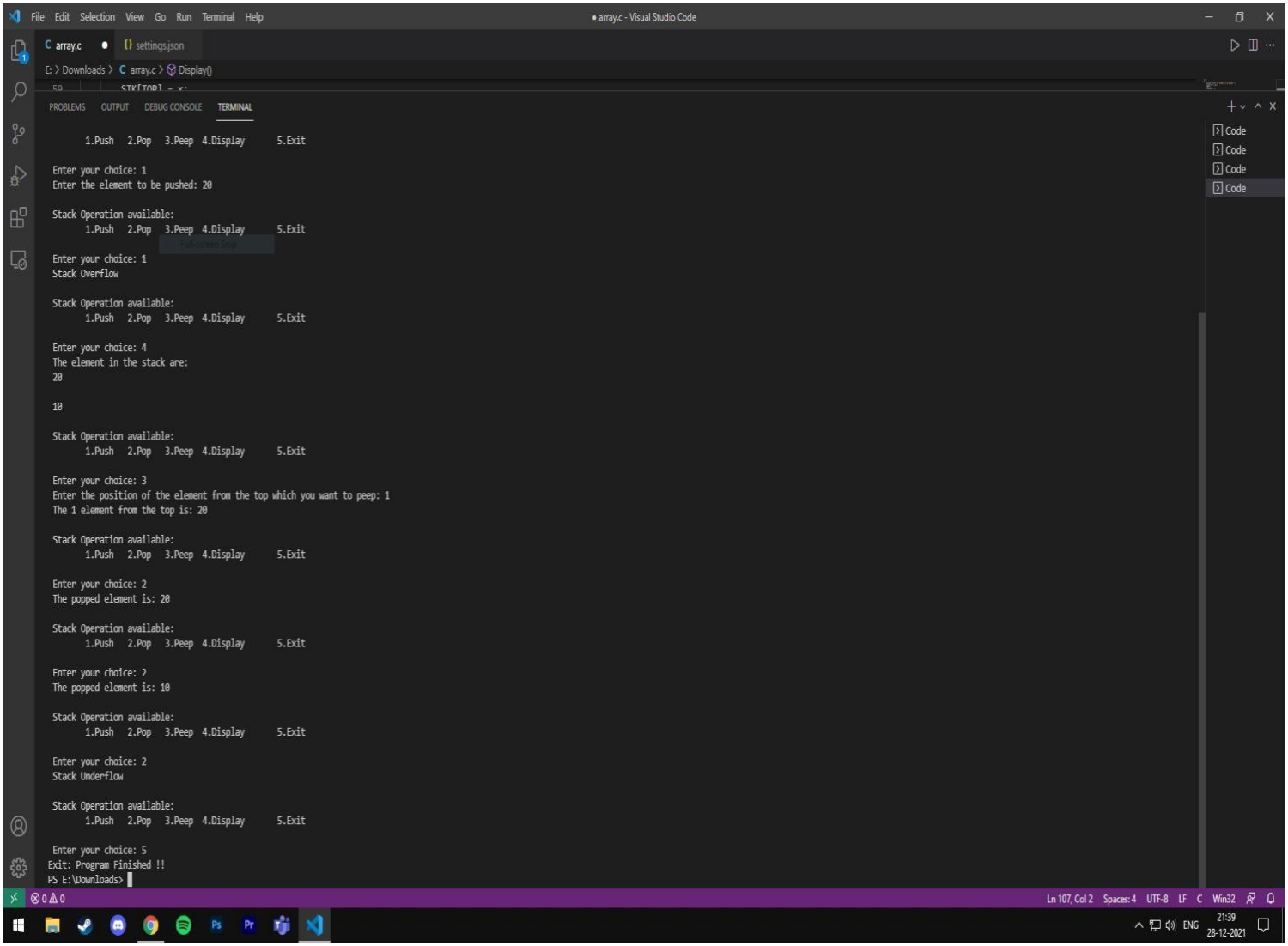

File Edit Selection View Go Run Terminal Help
• array.c - Visual Studio Code
array.c
settings.json
E: > Downloads > array.c > Display()
// Function to perform PUSH Operation
void Push()
{
if (TOP >= n - 1)
{
printf(" Stack Overflow \n");
}
else
{
printf(" Enter the element to be pushed: ");
scanf("%d", &x);
TOP++;
STK[TOP] = x;
}
}
// Function to perform POP Operation
void Pop()
{
if (TOP < 0)
{
printf(" Stack Underflow \n");
}
else
{
printf(" The popped element is: %d \n", STK[TOP]);
TOP--;
}
}
// Function to perform PEEP Opeartion
void Peep()
{
printf(" Enter the position of the element from the top which you want to peep: ");
scanf("%d", &i);
if (TOP - i + 1 < 0)
{
printf(" Stack Underflow on Peep \n");
}
else
{
printf(" The %d element from the top is: %d \n", i, STK[TOP - i + 1]);
}
}
// Function to DISPLAY the Stack
void Display()
{
Ln 107, Col 2 Spaces: 4 UTF-8 LF C Win32
ENG 21:37 28-12-2021



```
        STK[TOP] = x;
    }
}

// Function to perform POP Operation
void Pop()
{
    if (TOP < 0)
    {
        printf(" Stack Underflow \n");
    }
    else
    {
        printf(" The popped element is: %d \n", STK[TOP]);
        TOP--;
    }
}

// Function to perform PEEP Opeartion
void Peep()
{
    printf(" Enter the position of the element from the top which you want to peep: ");
    scanf("%d", &i);
    if (TOP - i + 1 < 0)
    {
        printf(" Stack Underflow on Peep \n");
    }
    else
    {
        printf(" The %d element from the top is: %d \n", i, STK[TOP - i + 1]);
    }
}

// Function to DISPLAY the Stack
void Display()
{
    if (TOP < 0)
    {
        printf(" Stack is empty \n");
    }
    else
    {
        printf(" The element in the stack are:");
        for (i = TOP; i > -1; i--)
        {
            printf("\n %d \n", STK[i]);
        }
    }
}
```



```
       WELCOME to Implementation of STACK using array !!
Enter the size of Stack (Maximum size = 100): 2

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 1
 Enter the element to be pushed: 10

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 1
 Enter the element to be pushed: 20

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 1
 Stack Overflow

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 4
 The element in the stack are:
 20

 10

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 3
 Enter the position of the element from the top which you want to peep: 1
 The 1 element from the top is: 20

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 2
 The popped element is: 20

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit

 Enter your choice: 2
 The popped element is: 10

 Stack Operation available:
       1.Push   2.Pop   3.Peep  4.Display      5.Exit
```


File Edit Selection View Go Run Terminal Help
• array.c - Visual Studio Code
array.c
settings.json
E: > Downloads > array.c > Display()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 1
Enter the element to be pushed: 20
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 1
Stack Overflow
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 4
The element in the stack are:
20
10
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 3
Enter the position of the element from the top which you want to peep: 1
The 1 element from the top is: 20
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 2
The popped element is: 20
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 2
The popped element is: 10
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 2
Stack Underflow
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 5
Exit: Program Finished !!
PS E:\Downloads>
Code
Code
Code
Code
Ln 107, Col 2 Spaces: 4 UTF-8 LF C Win32
ENG
21:39
28-12-2021