Indian Institute of Engineering Science and Technology, Shibpur

Department of Information Technology

Data Structure Laboratory 2021

Assignment – 3

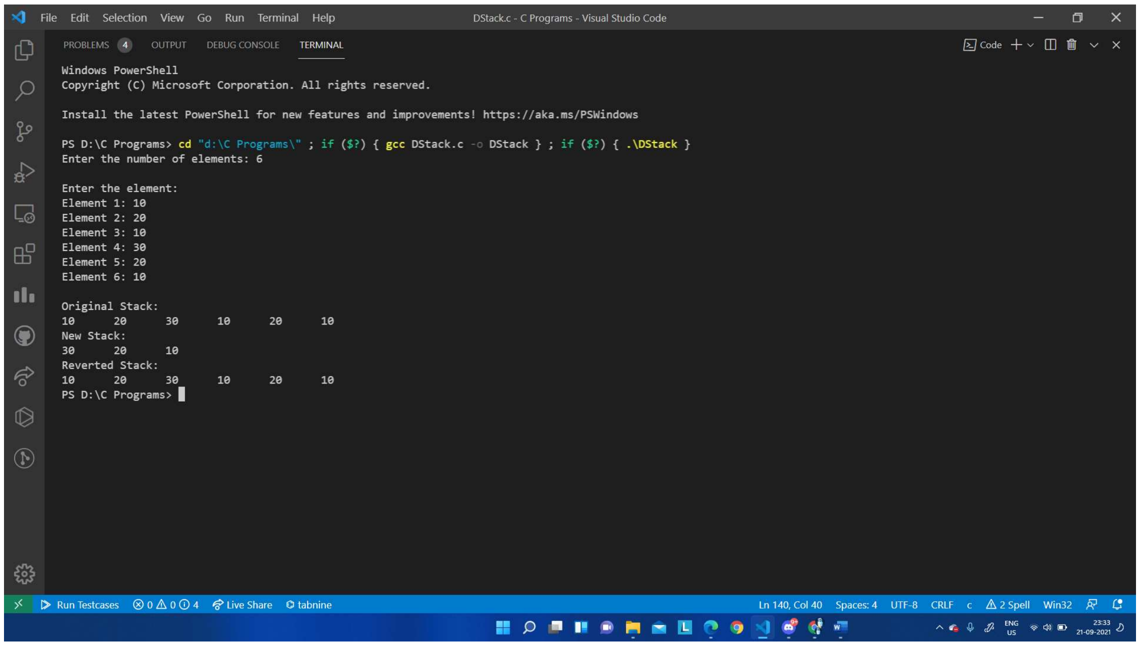
Tathagata Ghosh --- 2020ITB065 --- HY

[https://github.com/Tathagata-Ghosh-Developer/Lab-Assignment-3rd-Semester]

```c
/* Data Structure Lab Assignment 2021
Tathagata Ghosh --- 2020ITB065 ---- HY
21/09/2021 */

/*Q1. Write a function duplicateStack() that returns a new stack containing un
ique
elements in the same order as the given stack specified in the parameter.
Example:
Suppose the stack contains the elements : 10, 12, 10, 15, 20, 10, 15, 20, 25
So the duplicate stack should contain the elements: 10, 12, 15, 20, 25
Once this stack is created, display the contents of the stack and form this
stack revert back to the original stack contents.*/

#include <stdio.h>
#include <stdlib.h>

struct Stack
{
    int size, top, *arr;
};

struct Stack *setStack()
{
    struct Stack *ptr = (struct Stack *)malloc(sizeof(struct Stack));
    ptr->size = 100;
    ptr->top = -1;
    ptr->arr = (int *)malloc(ptr->size * sizeof(int));
    return ptr;
}

int isEmpty(struct Stack *ptr)
{
    if (ptr->top == -1)
        return 1;
    return 0;
}

int isFull(struct Stack *ptr)
{
    if (ptr->top == ptr->size - 1)
        return 1;
    return 0;
}

void push(struct Stack *ptr, int val)
{
    if (isFull(ptr))
    {
        printf("Stack Overflow!\n");
    }
    else
    {
        ptr->top++;
        ptr->arr[ptr->top] = val;
    }
}

int pop(struct Stack *ptr)
{
    if (isEmpty(ptr))
    {
        printf("Stack Underflow! 101");
        return -1;
    }
    else
    {
        int val = ptr->arr[ptr->top];
        ptr->top--;
        return val;
    }
}

void display(struct Stack *ptr)
{
    for (int i = ptr->top; i >= 0; i--)
    {
        printf("%d\t", ptr->arr[i]);
    }
    printf("\n");
}

int *container, length;

struct Stack *duplicateStack(struct Stack *ptr)
{
    struct Stack *new = setStack();
    length = ptr->top + 1;
    int arr[length];
    container = (int *)malloc(length * sizeof(int));

    for (int index = length - 1; !isEmpty(ptr); index--)
    {
        arr[index] = pop(ptr);
    }

    for (int index = 0; index < length; index++)
    {
        int var = 1;
        for (int i = 0; i <= new->top; i++)
        {
            if (arr[i] == arr[index])
            {
                container[index] = i;
                var = 0;
                break;
            }
        }
        if (var)
        {
            push(new, arr[index]);
            container[index] = new->top;
            arr[new->top] = arr[index];
        }
    }
    return new;
}

struct Stack *revertBack(struct Stack *ptr)
{
    struct Stack *reverted = setStack();
    int newlength = ptr->top + 1;
    int arr[newlength];

    for (int index = newlength - 1; !isEmpty(ptr); index--)
    {
        arr[index] = pop(ptr);
    }

    for (int index = 0; index < length; index++)
    {
        push(reverted, arr[container[index]]);
    }
    return reverted;
}
```

```c
int main()
{
    struct Stack *originalStack = setStack();
    int contentNum, content;

    printf("Enter the number of elements: ");
    scanf("%d", &contentNum);
    printf("\nEnter the element:\n");
    for (int i = 0; i < contentNum; i++)
    {
        printf("Element %d: ",i+1);
        scanf("%d", &content);
        push(originalStack, content);
    }
    printf("\n");

    printf("Original Stack:\n");
    display(originalStack);

    struct Stack *newStack = duplicateStack(originalStack);

    printf("New Stack:\n");
    display(newStack);

    struct Stack *revertedStack = revertBack(newStack);

    printf("Reverted Stack:\n");
    display(revertedStack);

    return 0;
}
```

OUTPUT:



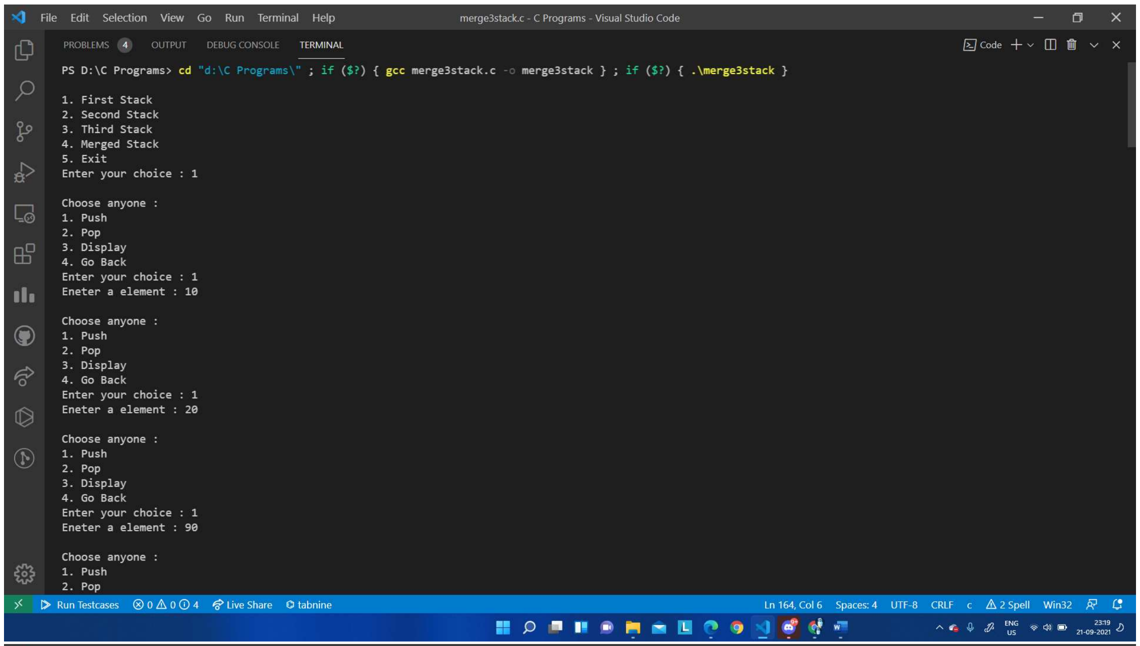
File Edit Selection View Go Run Terminal Help
DStack.c - C Programs - Visual Studio Code
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
Run Testcases
Live Share
tabnine
Ln 140, Col 40 Spaces: 4 UTF-8 CRLF c 2 Spell Win32

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc DStack.c -o DStack } ; if ($?) { .\DStack }
Enter the number of elements: 6

Enter the element:
Element 1: 10
Element 2: 20
Element 3: 10
Element 4: 30
Element 5: 20
Element 6: 10

Original Stack:
10      20      30      10      20      10
New Stack:
30      20      10
Reverted Stack:
10      20      30      10      20      10
PS D:\C Programs> 
```

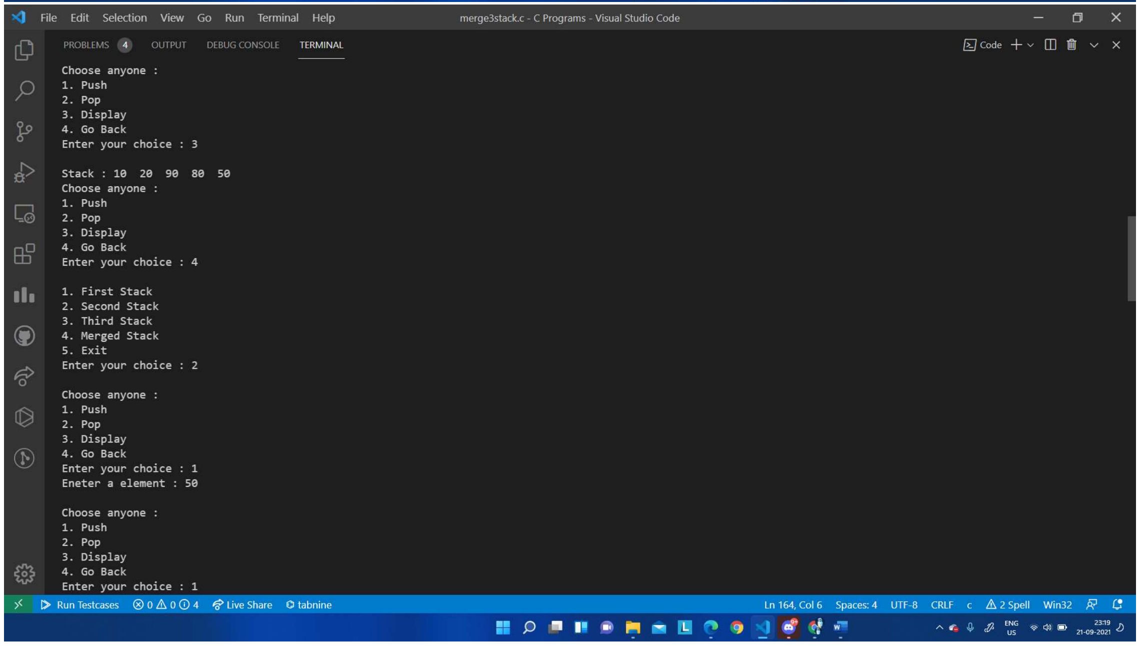
```c
/* Data Structure Lab Assignment 2021
Tathagata Ghosh --- 2020ITB065 ---- HY
21/09/2021 */

/*Q2. Sort the given set of elements present in three stack data structures an
d put
them in a single stack, without using any other variable.*/

#include <stdio.h>

void push(int Stack[], int element, int *top)
{
    Stack[*top] = element;
    *top += 1;
}
void pop(int Stack[], int *top)
{

    if (*top == 0)
    {
        printf("\nStack is Empty.");
    }
    else
    {
        *top -= 1;
        printf("\nPopped Element : %d", Stack[*top]);
    }
}
void displayStack(int Stack[], int top)
```

```c
{
    for (int i = 0; i < top; i++)
    {
        printf("%d  ", Stack[i]);
    }
}
void sort(int Stack[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (Stack[j] > Stack[j + 1])
            {
                int temp = Stack[j];
                Stack[j] = Stack[j + 1];
                Stack[j + 1] = temp;
            }
        }
    }
}
void merge(int Stack1[], int Stack2[], int Stack3[], int MergedStack[], int to
p1, int top2, int top3)
{
    sort(Stack1, top1);
    sort(Stack2, top2);
    sort(Stack3, top3);
    int i = 0, j = 0, k = 0, len=0;
    while (i < top1 && j < top2 && k < top3)
    {
        if (Stack1[i] < Stack2[j] && Stack1[i] < Stack3[k])
        {
            MergedStack[len++] = Stack1[i++];
        }
        else if (Stack2[j] < Stack3[k])
        {
            MergedStack[len++] = Stack2[j++];
        }
        else
        {
            MergedStack[len++] = Stack3[k++];
        }
    }
    while (i < top1 && j < top2)
    {
        if (Stack1[i] < Stack2[j])
        {
            MergedStack[len++] = Stack1[i++];
        }
        else
        {
            MergedStack[len++] = Stack2[j++];
        }
    }
    while (k < top3 && j < top2)
    {
        if (Stack2[j] < Stack3[k])
        {
            MergedStack[len++] = Stack2[j++];
        }
        else
        {
            MergedStack[len++] = Stack3[k++];
        }
    }
    while (i < top1 && k < top3)
    {
        if (Stack1[i] < Stack3[k])
        {
            MergedStack[len++] = Stack1[i++];
        }
        else
        {
            MergedStack[len++] = Stack3[k++];
        }
    }
    for (; i < top1; i++)
    {
        MergedStack[len++] = Stack1[i];
    }
    for (; j < top2; j++)
    {
        MergedStack[len++] = Stack2[j];
    }
    for (; k < top3; k++)
    {
        MergedStack[len++] = Stack3[k];
    }
}
int main()
{
    int Stack[4][100], top[4] = {0}, ch, ele, n;
    fflush(stdin);
    while (1)
    {
        printf("\n1. First Stack");
        printf("\n2. Second Stack");
        printf("\n3. Third Stack");
        printf("\n4. Merged Stack");
        printf("\n5. Exit");
        printf("\nEnter your choice : ");
        scanf("%d", &n);
        if (n > 4 || n < 1)
        {
            break;
        }
        if (n == 4)
        {
            merge(Stack[0], Stack[1], Stack[2], Stack[3], top[0], top[1], top[
2]);
            top[3] = top[0] + top[1] + top[2];
        }
        do
        {
            printf("\nChoose anyone :");
            printf("\n1. Push");
            printf("\n2. Pop");
            printf("\n3. Display");
            printf("\n4. Go Back");
            printf("\nEnter your choice : ");
            scanf("%d", &ch);
            switch (ch)
            {
            case 1:
                printf("Eneter a element : ");
                scanf("%d", &ele);
                push(Stack[n - 1], ele, &top[n - 1]);
                break;
            case 2:
                pop(Stack[n - 1], &top[n - 1]);
                break;
            case 3:
                printf("\nStack : ");
                displayStack(Stack[n - 1], top[n - 1]);
                break;
            }
        } while (ch != 4);
    }
    return 0;
}
```

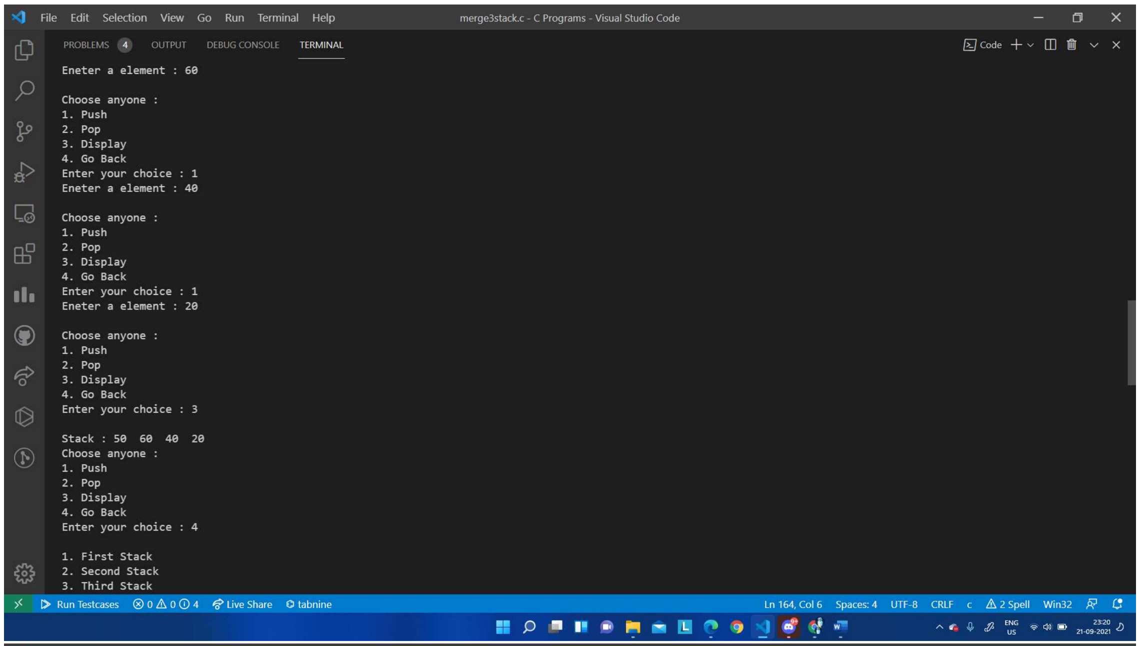
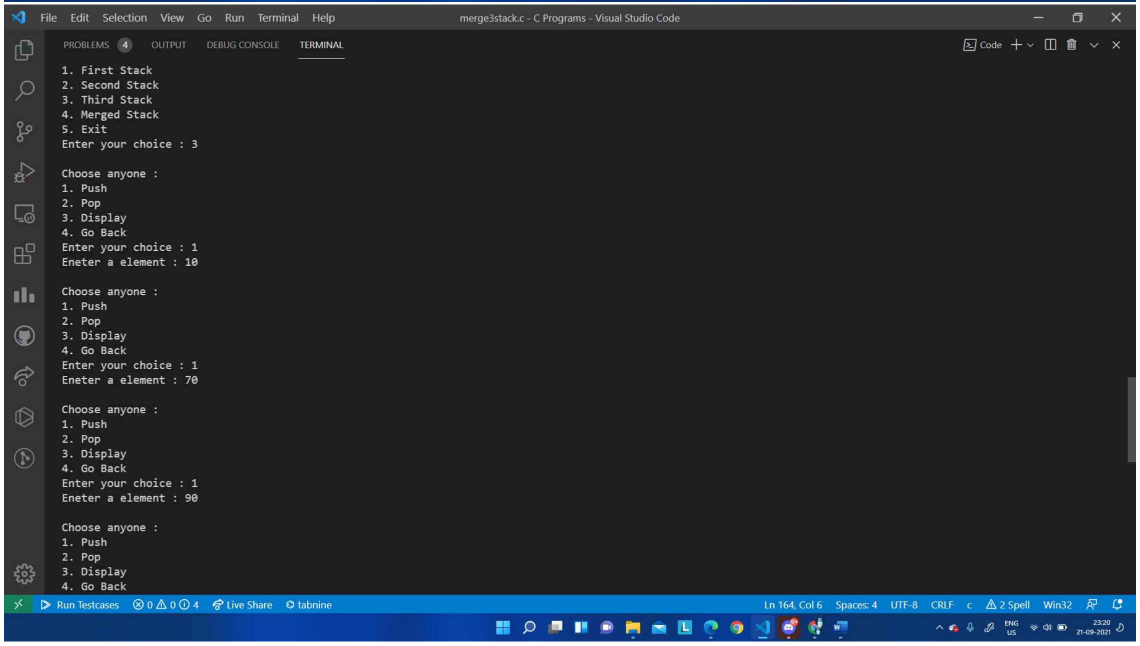
OUTPUT:

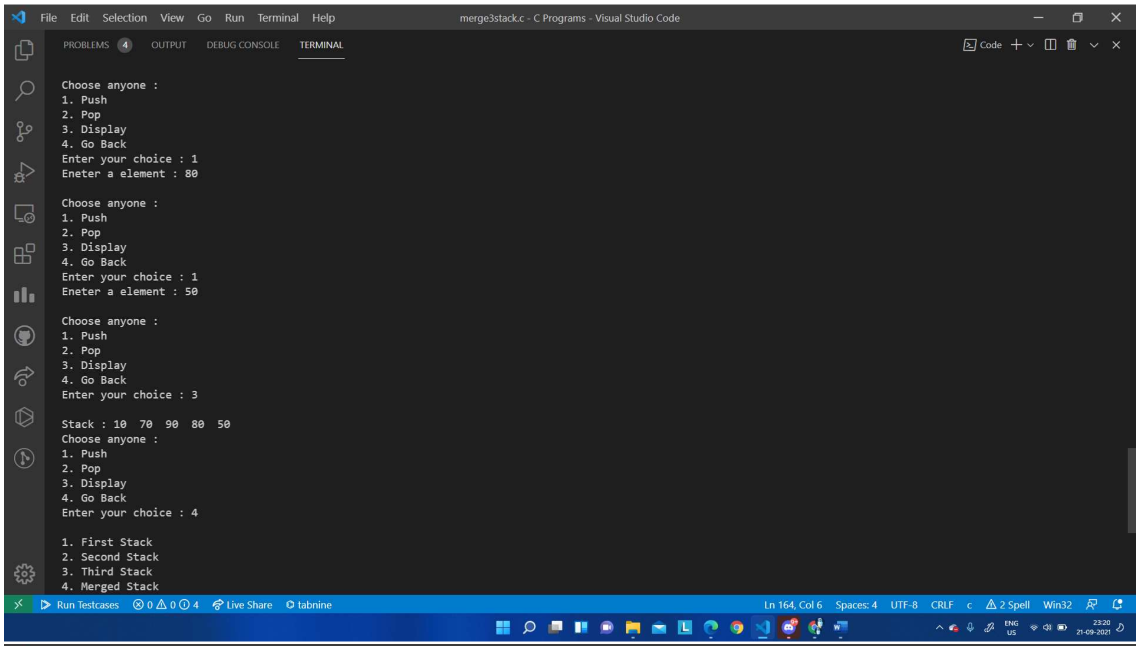


```
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc merge3stack.c -o merge3stack } ; if ($?) { .\merge3stack }

1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
5. Exit
Enter your choice : 1

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 10

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 20

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 90

Choose anyone :
1. Push
2. Pop
```

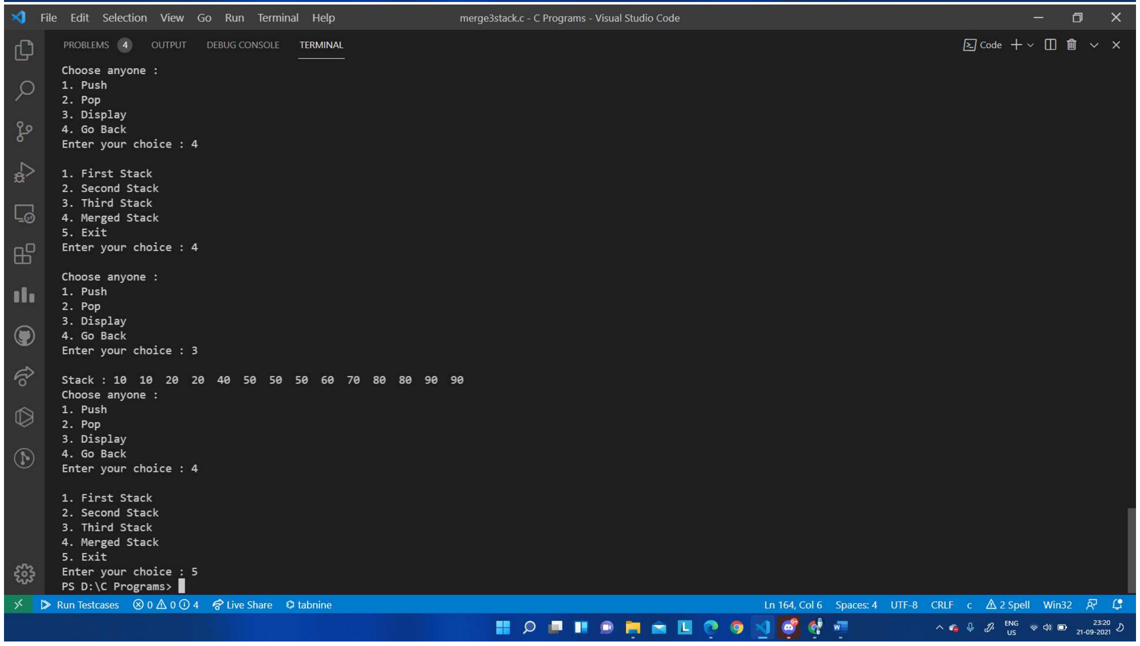


```
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 3

Stack : 10  20  90  80  50
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 4

1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
5. Exit
Enter your choice : 2

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 50

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
```


```
Eneter a element : 60

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 40

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 20

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 3

Stack : 50  60  40  20
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 4

1. First Stack
2. Second Stack
3. Third Stack
```
File Edit Selection View Go Run Terminal Help
merge3stack.c - C Programs - Visual Studio Code
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
Run Testcases
Live Share
tabnine
Ln 164, Col 6 Spaces: 4 UTF-8 CRLF c 2 Spell Win32

```
1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
5. Exit
Enter your choice : 3

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 10

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 70

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 90

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
```
File Edit Selection View Go Run Terminal Help
merge3stack.c - C Programs - Visual Studio Code
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
Run Testcases
Live Share
tabnine
Ln 164, Col 6 Spaces: 4 UTF-8 CRLF c 2 Spell Win32



```
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 80

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 1
Eneter a element : 50

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 3

Stack : 10  70  90  80  50
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 4

1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
```



```
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 4

1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
5. Exit
Enter your choice : 4

Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 3

Stack : 10  10  20  20  40  50  50  50  60  70  80  80  90  90
Choose anyone :
1. Push
2. Pop
3. Display
4. Go Back
Enter your choice : 4

1. First Stack
2. Second Stack
3. Third Stack
4. Merged Stack
5. Exit
Enter your choice : 5
PS D:\C Programs>
```