

**Lab No: 1 Date: 2081/**

**Title: Write a program to demonstrate Dynamic memory allocation in C.**

Sometimes the size of the array you declared may be insufficient. To solve this issue, you can allocate memory manually during run-time. Dynamic memory allocation in C is when memory allocated can be determined at run-time. There are 4 library functions provided by C defined under <stdlib.h> header file to facilitate dynamic memory allocation in C programming. They are:

1. Malloc()

The name "malloc" stands for memory allocation. The malloc() function reserves a block of memory of the specified number of bytes. And, it returns a pointer of void which can be casted into pointers of any form.

1. Calloc()

The name "calloc" stands for contiguous allocation. The malloc() function allocates memory and leaves the memory uninitialized, whereas the calloc() function allocates memory and initializes all bits to zero.

1. Free()

Dynamically allocated memory created with either calloc() or malloc() doesn't get freed on their own. You must explicitly use free() to release the space.

1. Realloc()

If the dynamically allocated memory is insufficient or more than required, you can change the size of previously allocated memory using the realloc() function.

**IDE: Visual Studio Code**

**Langauage: C**

**Source code (for Malloc):**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n, i, sum = 0;

    printf("Enter the value for n: ");

    scanf("%d", &n);

    int \*ptr;

    ptr = (int \*)malloc(n \* sizeof(int));

    printf("Before initilizing \n");

    for (i = 0; i < n; i++)

    {

        printf("%d", \*(ptr + i));

        printf("\n");

    }

    printf("\nAfter initilizing \n");

    printf("Enter %d values: \n", n);

    for (i = 0; i < n; i++)

    {

        scanf("%d", ptr + i);

        sum = sum + \*(ptr + i);

    }

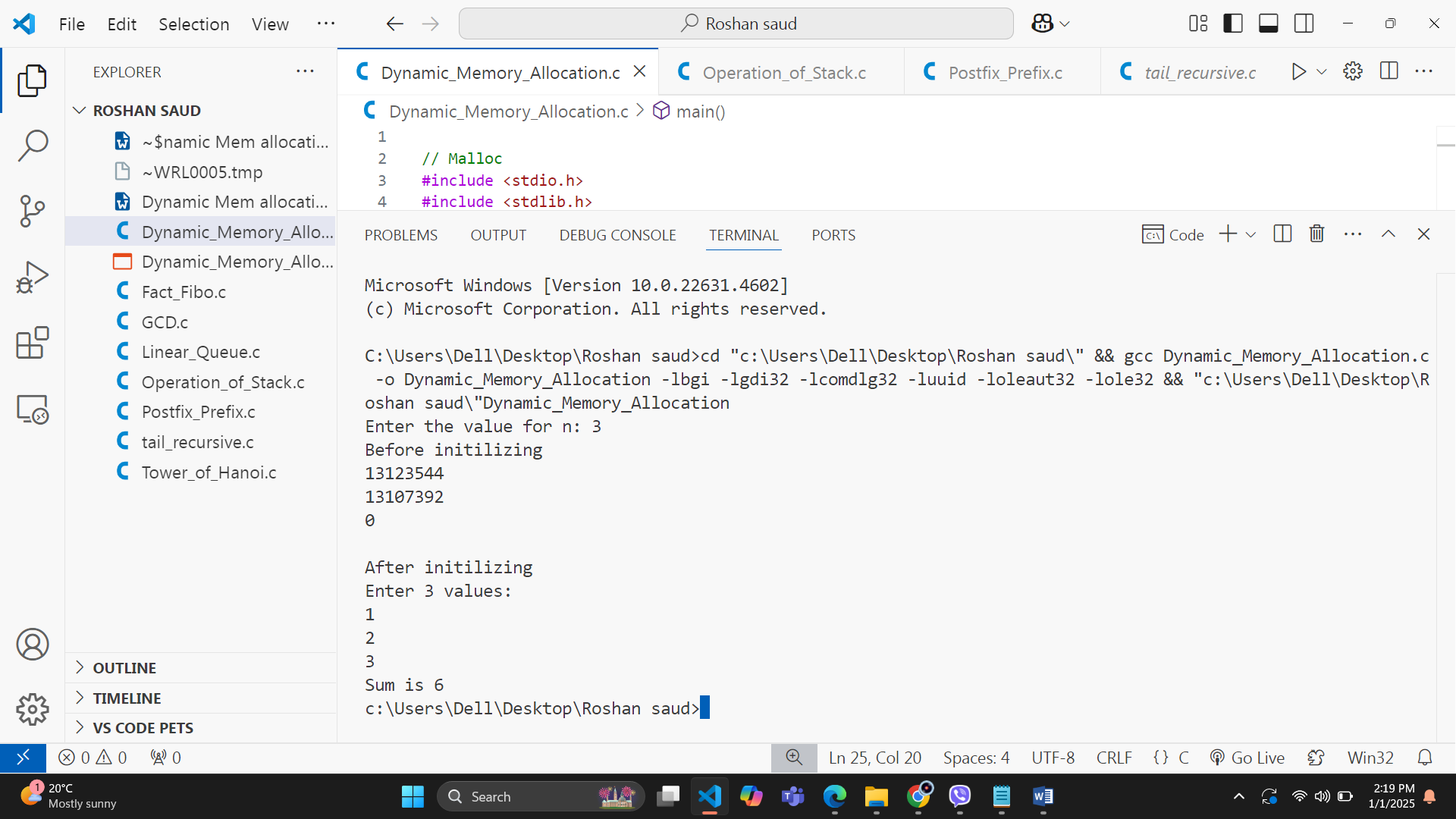
    printf("Sum is %d", sum);

    free(ptr);

    return 0;

}

**Output:**

****

**Source code (for Calloc):**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n, i, sum = 0;

    printf("Enter the value for n: ");

    scanf("%d", &n);

    int \*ptr;

    ptr = (int \*)calloc(n, 5 \* sizeof(int));

    printf("Before initilizing \n");

    for (i = 0; i < n; i++)

    {

        printf("%d", \*(ptr + i));

        printf("\n");

    }

    printf("\nAfter initilizing \n");

    printf("Enter %d values: \n", n);

    for (i = 0; i < n; i++)

    {

        scanf("%d", ptr + i);

        sum = sum + \*(ptr + i);

    }

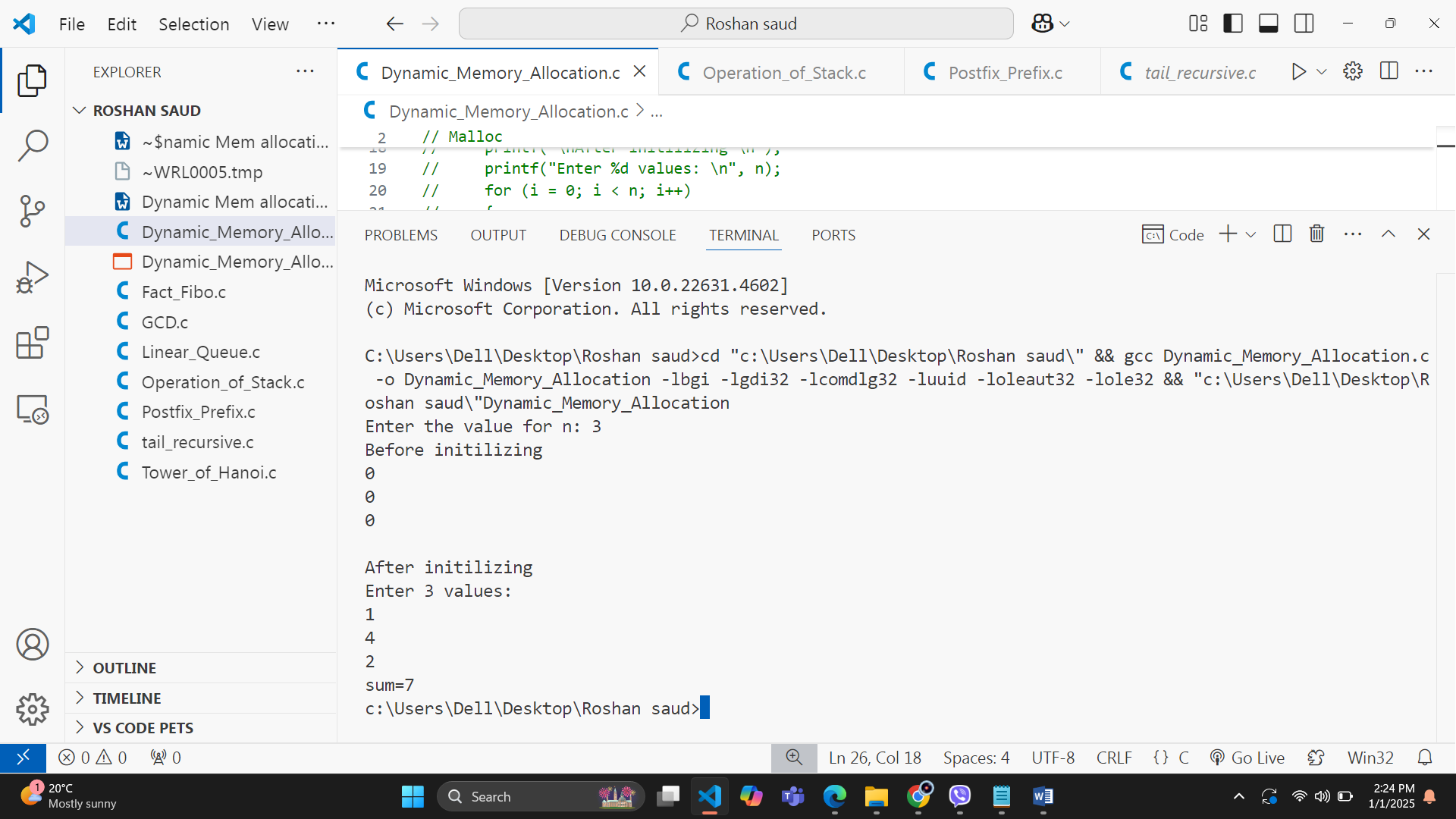
    printf("sum=%d", sum);

    free(ptr);

    return 0;

}

**Output:**

****

**Source code (For realloc):**

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n, n1, i, sum = 0;

    printf("Enter the value for n: ");

    scanf("%d", &n);

    int \*ptr;

    ptr = (int \*)malloc(n \* sizeof(int));

    printf("Before initilizing \n");

    for (i = 0; i < n; i++)

    {

        printf("%d", \*(ptr + i));

        printf("\n");

    }

    printf("\nAfter initilizing \n");

    printf("Enter %d values: \n", n);

    for (i = 0; i < n; i++)

    {

        scanf("%d", ptr + i);

        sum = sum + \*(ptr + i);

    }

    printf("Sum is %d\n", sum);

    printf("Enter number of additional size: ");

    scanf("%d", &n1);

    realloc(ptr, n1 \* sizeof(int));

    for (i = 0; i < n1; i++)

    {

        scanf("%d", ptr + i);

        sum = sum + \*(ptr + i);

    }

    printf("Additional numbers are:\n");

    for (i = 0; i < n1; i++)

    {

        printf("%d\n", \*(ptr + i));

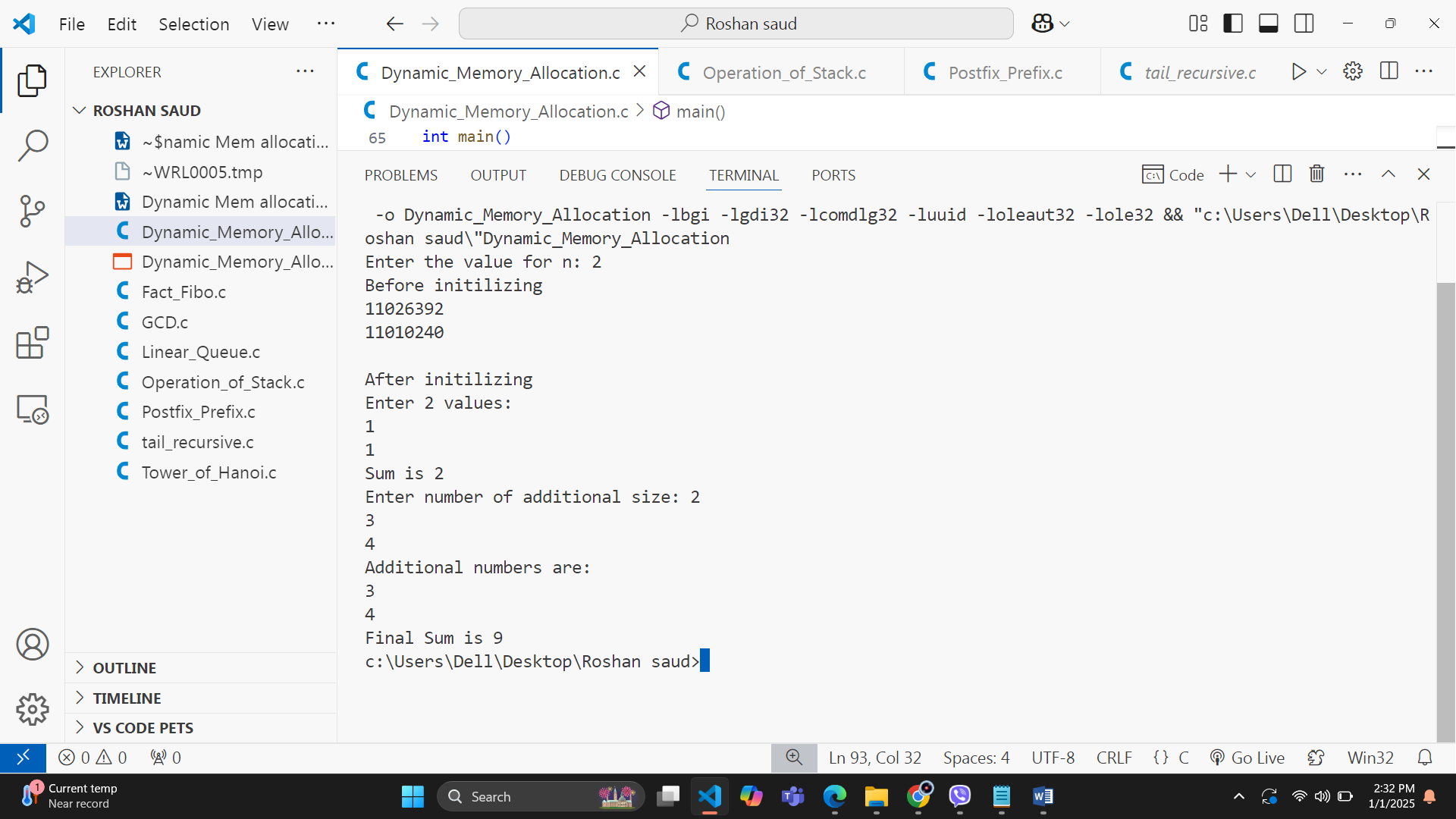
    }

    printf("Final Sum is %d", sum);

    return 0;

}

**Output:**

****