DS PRACTICAL 2

A8_B3_42

AMBER SHUKLA

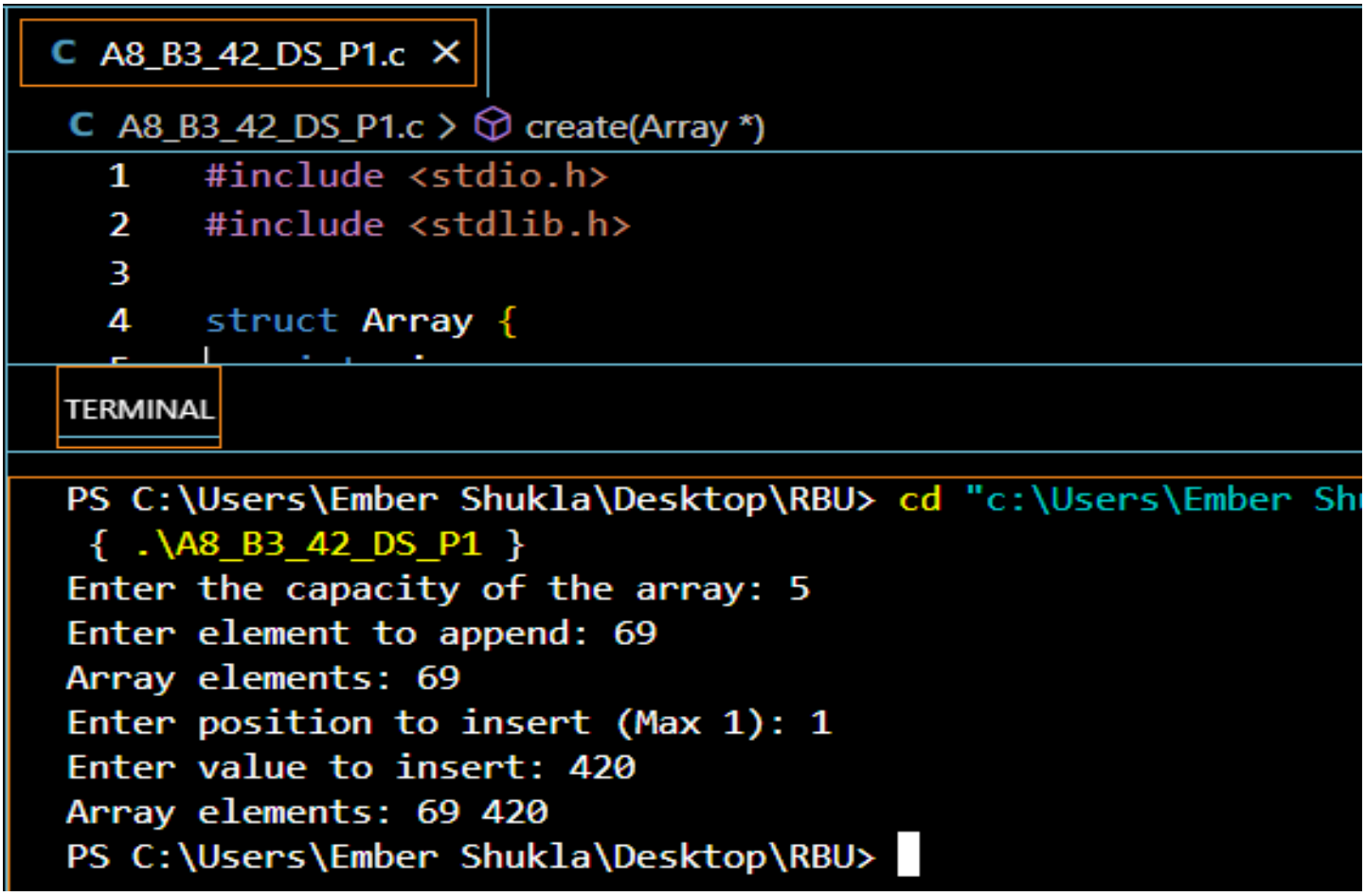
A8_B3_42_DS_P1.c

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

struct Array {
    int size;
    int length;
    int *a;
};

void create(struct Array *arr) {
    int capacity;
    printf("Enter the capacity of the array: ");
    scanf("%d", &capacity);
    arr->size = capacity;
    arr->length = 0;
    arr->a = (int *)malloc(arr->size * sizeof(int));
}
```

```c
void append(struct Array *arr) {
    int element;
    printf("Enter element to append: ");
    scanf("%d", &element);
    if (arr->length < arr->size) {
        arr->a[arr->length] = element;
        arr->length++;
    } else {
        printf("Array is full.\n");
    }
}

void display(struct Array *arr) {
    printf("Array elements: ");
    for (int i = 0; i < arr->length; i++) {
        printf("%d ", arr->a[i]);
    }
    printf("\n");
}
```

```c
void insert(struct Array *arr) {
    int pos, value;
    printf("Enter position to insert (Max %d): ", arr->length);
    scanf("%d", &pos);
    if (pos > arr->length) {
        printf("Invalid position.\n");
        return;
    }
    printf("Enter value to insert: ");
    scanf("%d", &value);
    for (int i = arr->length; i > pos; i--)
        arr->a[i] = arr->a[i-1];

    arr->a[pos] = value;
    arr->length++;
}
```

```c
int main() {
    struct Array arr;
    create(&arr);
    if (arr.a == NULL) {
        printf("Error\n");
        return 0;
    }
    append(&arr);
    display(&arr);
    insert(&arr);
    display(&arr);

    return 0;
}
```


A8_B3_42_DS_P1.c
A8_B3_42_DS_P1.c > create(Array *)
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 struct Array {
TERMINAL
PS C:\Users\Ember Shukla\Desktop\RBU> cd "c:\Users\Ember Sh
{ .\A8_B3_42_DS_P1 }
Enter the capacity of the array: 5
Enter element to append: 69
Array elements: 69
Enter position to insert (Max 1): 1
Enter value to insert: 420
Array elements: 69 420
PS C:\Users\Ember Shukla\Desktop\RBU>