

Name: Utkarsh Yadav

Roll: 23053172

DS Assignment for 01/08/2024

# Add and multiply 3 complex numbers using the old function

```
Learning C 3rd Sem - complexNum.c

1  #include <stdio.h>
2
3
4  typedef struct {
5      int real;
6      int imag;
7  } Complex;
8
9
10 Complex add(Complex a, Complex b) {
11     Complex result;
12     result.real = a.real + b.real;
13     result.imag = a.imag + b.imag;
14     return result;
15 }
16
17
18 Complex multiply(Complex a, Complex b) {
19     Complex result;
20     result.real = a.real * b.real - a.imag * b.imag;
21     result.imag = a.real * b.imag + a.imag * b.real;
22     return result;
23 }
24
25 int main() {
26     Complex num1, num2, num3;
27     Complex sumOne, productOne, sumTwo, productTwo;
28
29
30     printf("Enter the first complex number: ");
31     scanf("%d %d", &num1.real, &num1.imag);
32
33
34     printf("Enter the second complex number: ");
35     scanf("%d %d", &num2.real, &num2.imag);
36
37     printf("Enter the third complex number: ");
38     scanf("%d %d", &num3.real, &num3.imag);
39
40     sumOne = add(num1, num2);
41     productOne = multiply(num1, num2);
42     sumTwo = add(sumOne, num3);
43     productTwo = multiply(productOne, num3);
44
45     printf("Initial Sum: %d + %di\n", sumOne.real, sumOne.imag);
46     printf("Initial Product: %d + %di\n", productOne.real, productOne.imag);
47
48
49
50     printf("Final Sum: %d + %di\n", sumTwo.real, sumTwo.imag);
51     printf("Final Product: %d + %di\n", productTwo.real, productTwo.imag);
52
53
54
55     return 0;
56 }
57
```

KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment\_aug1

• \$ ./cn

Enter the first complex number: 2 3

Enter the second complex number: 4 5

Enter the third complex number: 6 7

Initial Sum:  $6 + 8i$

Initial Product:  $-7 + 22i$

Final Sum:  $12 + 15i$

Final Product:  $-196 + 83i$

## Return a struct pointer in the same program

Learning C 3rd Sem - cnReturnPointer.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct {
5     int real;
6     int imag;
7 } Complex;
8
9 Complex* add(Complex a, Complex b) {
10     Complex* result = (Complex*) malloc(sizeof(Complex));
11     (*result).real = a.real + b.real;
12     (*result).imag = a.imag + b.imag;
13     return result;
14 }
15
16 Complex* multiply(Complex a, Complex b) {
17     Complex* result = (Complex*) malloc(sizeof(Complex));
18     (*result).real = a.real * b.real - a.imag * b.imag;
19     (*result).imag = a.real * b.imag + a.imag * b.real;
20     return result;
21 }
22
23 int main() {
24     Complex num1, num2, num3;
25     Complex *sumOne, *productOne, *sumTwo, *productTwo;
26
27     printf("Enter the first complex number: ");
28     scanf("%d %d", &num1.real, &num1.imag);
29
30     printf("Enter the second complex number: ");
31     scanf("%d %d", &num2.real, &num2.imag);
32
33     printf("Enter the third complex number: ");
34     scanf("%d %d", &num3.real, &num3.imag);
35
36     sumOne = add(num1, num2);
37     productOne = multiply(num1, num2);
38     sumTwo = add(*sumOne, num3);
39     productTwo = multiply(*productOne, num3);
40
41     printf("Initial Sum: %d + %di\n", (*sumOne).real, (*sumOne).imag);
42     printf("Initial Product: %d + %di\n", (*productOne).real, (*productOne).imag);
43     printf("Final Sum: %d + %di\n", (*sumTwo).real, (*sumTwo).imag);
44     printf("Final Product: %d + %di\n", (*productTwo).real, (*productTwo).imag);
45
46
47     free(sumOne);
48     free(productOne);
49     free(sumTwo);
50     free(productTwo);
51
52     return 0;
53 }
54
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
$ ./cnReturnPointer
Enter the first complex number: 2 3
Enter the second complex number: 4 5
Enter the third complex number: 6 7
Initial Sum: 6 + 8i
Initial Product: -7 + 22i
Final Sum: 12 + 15i
Final Product: -196 + 83i
```

# Dynamically allocate and display a 2D Array

```
Learning C 3rd Sem - dynamic2dArray.c

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void createArray(int rows, int cols);
5
6  int main() {
7      int rows, cols;
8
9
10     printf("Enter the number of rows: ");
11     scanf("%d", &rows);
12     printf("Enter the number of columns: ");
13     scanf("%d", &cols);
14
15     createArray(rows, cols);
16
17     return 0;
18 }
19
20 void createArray(int rows, int cols) {
21     int i, j;
22
23     int **array = (int **)malloc(rows * sizeof(int *));
24
25     for (i = 0; i < rows; i++) {
26         array[i] = (int *)malloc(cols * sizeof(int));
27         if (array[i] == NULL) {
28             printf("Memory allocation failed\n");
29             for (j = 0; j < i; j++) {
30                 free(array[j]);
31             }
32             free(array);
33             exit(1);
34         }
35     }
36
37
38     printf("Enter elements of the array:\n");
39     for (i = 0; i < rows; i++) {
40         for (j = 0; j < cols; j++) {
41             printf("Element [%d][%d]: ", i, j);
42             scanf("%d", &array[i][j]);
43         }
44     }
45
46
47     printf("The array is:\n");
48     for (i = 0; i < rows; i++) {
49         for (j = 0; j < cols; j++) {
50             printf("%d ", array[i][j]);
51         }
52         printf("\n");
53     }
54
55
56     for (i = 0; i < rows; i++) {
57         free(array[i]);
58     }
59     free(array);
60 }
61
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./array
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements of the array:
Element [0][0]: 1
Element [0][1]: 2
Element [1][0]: 3
Element [1][1]: 4
The array is:
1 2
3 4
```

## Example of Direct Recursion

Learning C 3rd Sem - directRecurrsion.c

```
1  #include <stdio.h>
2
3  void printNumbersDirect(int n) {
4      if (n > 0) {
5          printf("%d\n", n);
6          printNumbersDirect(n - 1);
7      }
8  }
9
10 int main() {
11     int n = 5;
12     printf("Direct Recursion:\n");
13     printNumbersDirect(n);
14     return 0;
15 }
16
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./dr
Direct Recursion:
5
4
3
2
1
```



## Example of Indirect Recursion

Learning C 3rd Sem - indirectRecurssion.c

```
1  #include <stdio.h>
2
3  void functionA(int n);
4  void functionB(int n);
5
6  void functionA(int n) {
7      if (n > 0) {
8          printf("%d\n", n);
9          functionB(n - 1);
10     }
11 }
12
13
14 void functionB(int n) {
15     if (n > 0) {
16         printf("%d\n", n);
17         functionA(n - 1);
18     }
19 }
20
21 int main() {
22     int n = 5;
23     printf("Indirect Recursion:\n");
24     functionA(n);
25     return 0;
26 }
27
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./ir
Indirect Recursion:
5
4
3
2
1
```

## Malloc Example

Learning C 3rd Sem - mallocExample.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int *arr;
6      int n = 5;
7
8      arr = (int*) malloc(n * sizeof(int));
9
10     for (int i = 0; i < n; i++) {
11         arr[i] = i + 1;
12         printf("%d ", arr[i]);
13     }
14     printf("\n");
15
16     free(arr);
17
18     return 0;
19 }
20
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
• $ ./malloc
• 1 2 3 4 5
```

## Calloc Example

Learning C 3rd Sem - callocExample.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int *arr;
6      int n = 5;
7
8      arr = (int*) calloc(n, sizeof(int));
9
10     for (int i = 0; i < n; i++) {
11         printf("%d ", arr[i]);
12     }
13     printf("\n");
14
15     free(arr);
16
17     return 0;
18 }
19
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./calloc
○0 0 0 0 0
```

## Realloc Example

Learning C 3rd Sem - reallocExample.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      int *arr;
6      int n = 5, new_n = 10;
7
8      arr = (int*) malloc(n * sizeof(int));
9
10     for (int i = 0; i < n; i++) {
11         arr[i] = i + 1;
12     }
13
14     printf("Before realloc: ");
15
16     for (int i = 0; i < n; i++) {
17         printf("%d ", arr[i]);
18     }
19
20     printf("\n");
21
22     arr = (int*) realloc(arr, new_n * sizeof(int));
23
24     for (int i = n; i < new_n; i++) {
25         arr[i] = i + 1;
26     }
27
28     printf("After realloc: ");
29
30     for (int i = 0; i < new_n; i++) {
31         printf("%d ", arr[i]);
32     }
33
34     printf("\n");
35
36     free(arr);
37
38     return 0;
39 }
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./realloc
Before realloc: 1 2 3 4 5
After realloc: 1 2 3 4 5 6 7 8 9 10
```

## Validate the structure and union output from the ppt

Learning C 3rd Sem - validate.c

```
1  #include <stdio.h>
2
3  typedef struct {
4      int rollno;
5      int marks;
6  } studentStruct;
7
8  typedef union {
9      int rollno;
10     float marks;
11 } studentUnion;
12
13 int main() {
14
15     studentStruct s1;
16
17     s1.rollno = 20;
18     s1.marks = 90;
19
20     printf("Using Structure: \n");
21     printf("Roll Number: %d\n", s1.rollno);
22     printf("Marks: %d\n", s1.marks);
23
24
25     studentUnion s2;
26
27     printf("Using Union: \n");
28
29     s2.rollno = 20;
30     printf("Roll Number: %d\n", s2.rollno);
31
32     s2.marks = 90.0;
33     printf("Marks: %.2f\n", s2.marks);
34
35     printf("Roll Number after assigning marks: %d\n", s2.rollno);
36
37     return 0;
38 }
39
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
•$ ./validate
Using Structure:
Roll Number: 20
Marks: 90
Using Union:
Roll Number: 20
Marks: 90.00
Roll Number after assigning marks: 1119092736
```



## Struct to store and display Book details

Learning C 3rd Sem - bookStruct.c

```
1  #include <stdio.h>
2
3  typedef struct {
4      char name[100];
5      char author[50];
6      int bookID;
7      char pubYear[5];
8  } Book;
9
10 int main() {
11
12     Book book;
13
14     printf("Enter the book name: ");
15     scanf("%s", book.name);
16
17     printf("Enter the author's name: ");
18     scanf("%s", book.author);
19
20     printf("Enter the book ID: ");
21     scanf("%d", &book.bookID);
22
23     printf("Enter the publication year: ");
24     scanf("%s", book.pubYear);
25
26     printf("\nBook Details:\n");
27     printf("Name: %s\n", book.name);
28     printf("Author: %s\n", book.author);
29     printf("Book ID: %d\n", book.bookID);
30     printf("Publication Year: %s\n", book.pubYear);
31
32     return 0;
33 }
34
```

```
KIIT0001@Utkarsh MINGW64 /d/Learning C 3rd Sem/assignment_aug1
• $ ./book
Enter the book name: Eclipse
Enter the author's name: Meyer
Enter the book ID: 001
Enter the publication year: 2007

Book Details:
Name: Eclipse
Author: Meyer
Book ID: 1
Publication Year: 2007
```