

```

1  #include <stdio.h>
2
3  struct student{
4      int r_no;
5      char name[20];
6      char course[20];
7      int fees;
8  };
9
10 int main() {
11     struct student stud1,*ptr;
12     ptr=&stud1;
13     printf("Enter the details of the student:\nRoll Number: ");
14     scanf("%d",&ptr->r_no);
15     printf("Name: ");
16     scanf("%s",ptr->name);
17     printf("Course: ");
18     scanf("%s",ptr->course);
19     printf("Fee: ");
20     scanf("%d",&ptr->fees);
21     printf("\n\nDETAILS OF THE STUDENT\nROLL NUMBER: %d\n",ptr->r_no);
22     printf("NAME: %s\n",ptr->name);
23     printf("COURSE: %s\n",ptr->course);
24     printf("FEES: %d\n",ptr->fees);
25     return 0;

```

A function declaration without a p

```

Enter the details of the student:
Roll Number: 221011
Name: Siti
Course: Programming
Fee: 46.50

```

```

DETAILS OF THE STUDENT
ROLL NUMBER: 221011
NAME: Siti
COURSE: Programming
FEES: 46
Program ended with exit code: 0

```



```

1  #include <stdio.h>
2
3  typedef struct distance {
4      int kms;
5      int meters;
6  } DISTANCE;
7
8  DISTANCE addDistance(DISTANCE d1, DISTANCE d2);
9
10 int main() {
11     DISTANCE d1 = {0, 0}, d2 = {0, 0}, d3 = {0, 0};
12     int option;
13     int inputFlag = 0;
14
15     do {
16         // Display the main menu
17         printf("*****MAIN MENU*****\n");
18         printf("1. Input distances\n");
19         printf("2. Add the distances\n");
20         printf("3. EXIT\n");
21         printf("Enter your option: ");
22         scanf("%d", &option);
23         printf("\n");
24
25         switch (option) {
26             case 1:
27                 printf("Enter the first distance:\n");
28                 printf("Kilometers: ");
29                 scanf("%d", &d1.kms);
30                 printf("Meters: ");
31                 scanf("%d", &d1.meters);
32
33                 printf("Enter the second distance:\n");
34                 printf("Kilometers: ");
35                 scanf("%d", &d2.kms);
36                 printf("Meters: ");
37                 scanf("%d", &d2.meters);
38

```



A function declaration without a p

```

10 int main() {
38
39     inputFlag = 1;
40     printf("Distances input successfully!\n\n");
41     break;
42
43     case 2:
44         if (inputFlag) {
45             d3 = addDistance(d1, d2);
46             printf("The sum is %d kms and %d meters\n\n", d3.kms, d3.meters);
47         } else {
48             printf("Please input distances first (Option 1)!\n\n");
49         }
50         break;
51
52     case 3:
53         printf("Exiting the program. Goodbye!\n\n");
54         break;
55
56     default:
57         printf("Invalid option! Please try again.\n\n");
58         break;
59 }
60 while (option != 3);
61
62 return 0;
63 }
64
65 DISTANCE addDistance(DISTANCE d1, DISTANCE d2) {
66     DISTANCE result;
67     result.kms = d1.kms + d2.kms;
68     result.meters = d1.meters + d2.meters;
69
70     while (result.meters >= 1000) {
71         result.meters = result.meters % 1000;
72         result.kms += 1;
73     }
74
75     return result;
76 }
77

```

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 1

Enter the first distance:

Kilometers: 30

Meters: 21

Enter the second distance:

Kilometers: 34

Meters: 35

Distances input successfully!

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 2

The sum is 64 kms and 56 meters

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 5

Invalid option! Please try again.

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 3

Exiting the program. Goodbye!

Program ended with exit code: 0


```
1  #include <stdio.h>
2
3  typedef struct distance {
4      int kms;
5      int meters;
6  } DISTANCE;
7
8  DISTANCE addDistance(int kms1, int meters1, int kms2, int meters2);
9
10 int main() {
11     DISTANCE d1 = {0, 0}, d2 = {0, 0}, d3 = {0, 0};
12     int option;
13     int inputFlag = 0;
14
15     do {
16         printf("*****MAIN MENU*****\n");
17         printf("1. Input distances\n");
18         printf("2. Add the distances\n");
19         printf("3. EXIT\n");
20         printf("Enter your option: ");
21         scanf("%d", &option);
22         printf("\n");
23
24         switch (option) {
25             case 1:
26                 printf("Enter the first distance in kms and meters: ");
27                 scanf("%d %d", &d1.kms, &d1.meters);
28
29                 printf("Enter the second distance in kms and meters: ");
30                 scanf("%d %d", &d2.kms, &d2.meters);
31                 inputFlag = 1;
32                 printf("Distances input successfully!\n\n");
33                 break;
34
35             case 2:
36                 if (inputFlag) {
37                     d3 = addDistance(d1.kms, d1.meters, d2.kms, d2.meters);
```

```

10 int main() {
34
35     case 2:
36         if (inputFlag) {
37             d3 = addDistance(d1.kms, d1.meters, d2.kms, d2.meters);
38             printf("The sum is %d kms and %d meters\n\n", d3.kms, d3.meters);
39         } else {
40             printf("Please input distances first (Option 1)!\n\n");
41         }
42         break;
43
44     case 3:
45         printf("Exiting the program. Goodbye!\n\n");
46         break;
47
48     default:
49         printf("Invalid option! Please try again.\n\n");
50         break;
51 }
52 while (option != 3);
53
54 return 0;
55 }
56
57 DISTANCE addDistance(int a, int b, int c, int d) {
58     DISTANCE result;
59     result.kms = a + c;
60     result.meters = b + d;
61
62     if (result.meters >= 1000) {
63         result.kms += result.meters / 1000;
64         result.meters = result.meters % 1000;
65     }
66
67     return result;
68 }

```

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 4

Invalid option! Please try again.

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 2

Please input distances first (Option 1)!

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 1

Enter the first distance in kms and meters: 34

23

Enter the second distance in kms and meters: 24

24

Distances input successfully!

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 1

Enter the first distance in kms and meters: 34
23

Enter the second distance in kms and meters: 24
24

Distances input successfully!

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 2

The sum is 58 kms and 47 meters

*****MAIN MENU*****

1. Input distances
2. Add the distances
3. EXIT

Enter your option: 3

Exiting the program. Goodbye!

Program ended with exit code: 0

