

1) Create structure item details with members item\_name, quantity, price total amount. calculate party price.

#include <stdio.h>

#include <stdlib.h>

struct item\_details

{

char item\_name[20];

float quantity;

float price;

float total\_amount;

};

int main()

{

struct item\_details s1 = {"pen", 100.00, 360.00, 00.00};

printf("%.2f\n", s1.price);

printf("%.2f\n", s1.quantity);

printf(s1.item\_name);

s1.total\_amount = s1.price \* s1.quantity;

printf("%.2f", s1.total\_amount);

~~set s2~~

struct item\_details s2 = {"papad", 110.00, 100.00, 00.00};

s2.total\_amount = s2.price \* s2.quantity;

float total\_expense = s1.total\_amount + s2.total\_amount;

printf("total party expense: %.2f\n", total\_expense);

return 0;

}

output

total party expense: 36000.00

2) Create a structure with name student with structure member : name, usn, gradelist of sem1, sem2. The student will be promoted to 3rd semester if he/she is not having backlog, of credit count  $\geq 16$ .

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

```
struct student {  
    char name[20];  
    int usn;  
    int sem1[4];  
    int sem2[5];  
};
```

```
int main() {
```

```
    struct student s1 = {"aaa", 1001, {2, 2, 3, 1}, {2, 1, 1, 2}};
```

```
    int sum;
```

```
    for (int i=0; i<4; ++i)
```

```
    {  
        sum = sum + s1 s1.sem1[i] + s1.sem2[i];  
    }
```

```
    printf("name: %s\n", s1.name);
```

```
    printf("usn: %d\n", s1.usn);
```

```
    printf("total credit : %d\n", sum);
```

```
    if (sum < 16) {
```

```
        printf("the student is eligible for 3rd semester");
```

```
    }  
    else {
```

```
        printf("the student is not eligible for 3rd semester");
```

```
    }
```

```
    return 0;
```

```
}
```

*Sum*  
18/12/23

output:

name: aad

uin: 1001

totalcredit: 141

the student is eligible for 3rd semester

4) Given an array `arr[]` containing `N` distances of the 'inch-feet' system, such that each element of the array represents a 'inch-feet' distance using structure

ex: input: `arr[] = {{10, 3.7}, {15, 5.0}, {6, 8}};`

output: feet sum: 26 Inch sum: 17.2

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

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

```
void arrcalc(float arr[3][2])
```

```
{
```

```
    float feet_sum;
```

```
    float inch_sum;
```

```
    for (int i=0; i<3; ++i)
```

```
    {
```

```
        feet_sum = feet_sum + arr[i][0];
```

```
        inch_sum = inch_sum + arr[i][1];
```

```
    }
```

```
    printf("feet sum: %.2f\n", feet_sum);
```

```
    printf("inch sum: %.2f\n", inch_sum);
```

```
}
```

```
int main()
```

```
{
```

```
    float arr[3][2] = {{10.00, 3.70}, {15.00, 5.00}, {6.00, 8.00}};
```

```
    arrcalc(arr);
```

```
    return 0;
```

```
}
```

Output:

feet sum: 31.00

inch sum: 17.00

## pointer program

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

```
int main()
```

```
{
```

```
    int var = 10;
```

```
    int *ptr;
```

```
    ptr = &var;
```

```
    printf("var value = %d\n", var);
```

```
    printf("var address = %u\n", &var);
```

```
    printf("ptr = %u\n", ptr);
```

```
    printf("*ptr = %d\n", *ptr);
```

```
    printf("&ptr = %u\n", &ptr);
```

output:

var value = 10

var address = 6422300

ptr = 6422300

\*ptr = 10

&ptr = 6422296

## pointer and function program

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

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

```
void swap(int *x, int *y)
```

```
{
```

```
    int t;
```

```
    t = *x;
```

```
    *x = *y;
```

```
    *y = t;
```

```
}
```

```
int main()
```

```
{
```

```
    int n1, n2;
```

```
    printf("enter num1: ");
```

```
    scanf("%d", &n1);
```

```
    printf("enter num2: ");
```

```
    scanf("%d", &n2);
```

```

printf("before swapping : %d %d\n", n1, n2);
swap(&n1, &n2);
printf("after swapping : %d %d\n", n1, n2);
}

```

array - element insertion in a specified position and deletion in specified position, do this using functions.

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

```
int arr[5];
```

```
void insertarr(int i, int v)
```

```
{
    arr[i] = v;
}
```

```
void deletearr(int i)
```

```
{
    for (int j = 0; j < 5; j++)
    {
        printf("index: %d, value: %d\n", j, arr[j]);
    }
}
```

```
int main()
```

```
{
    insertarr(0, 61);
    insertarr(1, 95);
    insertarr(2, 62);
    displayarr();
    deletearr(2);
    displayarr();
}
```

output:

index 0, value: 61

index 1, value: 95

index 2, value: 62

index 3, value: 0

index 4, value: 0

index 0, value: 61

index 1, value: 95

index 2, value: 0

index 3, value: 0

index 4, value: 0



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct item_details
5  {
6      char item_name[20];
7      float quantity;
8      float price;
9      float total;
10 };
11
12 int main()
13 {
14     struct item_details s1 = {"paneer", 100.00, 250.00, 00.00};
15
16     s1.total = s1.price * s1.quantity;
17     // printf("%.2f", s1.total);
18
19     struct item_details s2 = {"papad", 110.00, 100.00, 00.00};
20     s2.total = s2.price * s2.quantity;
21
22     float total_expense = s1.total + s2.total;
23     printf("total party expenses: %.2f", total_expense);
24 }
```

PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL      PORTS

[Running] cd "d:\3rd\_sem\C learning\" && gcc 11.c

total party expenses:36000.00

[Done] exited with code=0 in 2.204 seconds



rd\_sem > C learning > C l2.c > ...

```
1  #include <stdio.h>
2
3  struct student
4  {
5      char name[20];
6      int usn;
7      int sem1[4];
8      int sem2[4];
9  };
10
11 int main()
12 {
13     struct student s1 = {"aaa", 1001, {2, 2, 3, 1}, {2, 1, 1, 2}};
14     int sum = 0;
15     for (int i = 0; i < 4; i++)
16     {
17         sum = sum + s1.sem1[i] + s1.sem2[i];
18     }
19     printf("name:%s\n", s1.name);
20     printf("usn:%d\n", s1.usn);
21     printf("total_credit:%d\n", sum);
22
23     if (sum < 16)
24     {
25         printf("the student is eligible for 3rd sem");
26     }
27     else
28     {
29         printf("the student is not eligible for 3rd sem");
30     }
31 }
```



```
[Running] cd "d:\3rd_sem\C learning\" && gcc 12.c
```

```
name:aaa
```

```
usn:1001
```

```
total_credit:14
```

```
the student is eligible for 3rd sem
```

```
[Done] exited with code=0 in 1.275 seconds
```

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 void arrcalc(float arr[3][2])
5 {
6     float feet_sum;
7     float inch_sum;
8     for(int i=0;i<3;++i)
9     {
10         feet_sum=feet_sum+arr[i][0];
11         inch_sum=inch_sum+arr[i][1];
12     }
13     printf("feetsum:%.2f\n",feet_sum);
14     printf("inchsum:%.2f\n",inch_sum);
15 }
16
17 int main()
18 {
19     float arr1[3][2]={{10.00,3.70},{15.00,5.50},{6.00,8.00}};
20     arrcalc(arr1);
21     /*float sup[3][2]={{2.00,6.62},{2.00,8.49},{1.00,2.95}};
22     arrcalc(sup);*/
23     return 0;
24 }
```

[Running] cd "d:\3rd\_sem\C learning\" &

feetsum:31.00

inchsum:17.20

[Done] exited with code=0 in 1.943 seconds



```
1  #include<stdio.h>
2
3  int main()
4  {
5      int var=10;
6      int *ptr;
7      ptr=&var;
8
9      printf("var value=%d\n",var);
10     printf("var address=%u\n",&var);
11     printf("ptr=%u\n",ptr);
12     printf("*ptr=%d\n",*ptr);
13     printf("&ptr=%u\n",&ptr);
14
15 }
```



[Running] cd "d:\3rd\_sem\C learning

var value=10

var address=6422300

ptr=6422300

\*ptr=10

&ptr=6422296

[Done] exited with code=0 in 1.424 s

```
1  #include<stdio.h>
2  int arr[5];
3
4  void insertarr(int i,int v)
5  {
6      arr[i]=v;
7  }
8
9  void deletearr(int i)
10 {
11     arr[i]=0;
12 }
13
14 void displayarr()
15 {
16     for (int i=0;i<5;i++)
17     {
18         printf("index: %d , value: %d\n",i,arr[i]);
19     }
20 }
21
22 int main()
23 {
24     insertarr(0,65);
25     insertarr(1,95);
26     insertarr(2,62);
27     displayarr();
28     deletearr(2);
29     displayarr();
30
31 }
```



[Running] cd d:\src\sem\c 1

index: 0 , value: 65

index: 1 , value: 95

index: 2 , value: 62

index: 3 , value: 0

index: 4 , value: 0

index: 0 , value: 65

index: 1 , value: 95

index: 2 , value: 0

index: 3 , value: 0

index: 4 , value: 0

[Done] exited with code=0 in