11/12/23

1) Create structure itemdetails with members item-name, quantity, price, total amount. Calculate party expenses.

→

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

~~int main() {~~
Struct itemdetails {
        char item-name [50];
        int quantity;
        int price;
};

int main() {
    struct item-details itt[50]
     int n;
    printf(" Enter no. of items whose details
            has to be entered: ");
    scanf("%d", &n);

   for(int i=0 ; i<n ; i= i+1) {
        printf ("Enter item name: ");
        scanf ("%s", &itt[i].name);
        print ("Enter quantity: ");
        scanf("%d", & itt[i].quantity);
        printf("Enter price per item: );"
        scanf("%d", & itt[i].price);
   }
int  sum=0 , p[50];
for (int  i=0 ; i<n ; i=i+1){
     p[i] = (it[i].quantity)*(it[i].price);
}

for (int i=0  ; i<n ; i=i+1){
          sum = sum + p[i];
}
printf("The total party expense = %d", sum);

return 0;
}
```

Output

```
Enter no. of items whose details has to be
   entered: 2
Enter item name: Speaker
 Enter quantity  : 2
Enter price per item: 1200
Enter item name: ~~chicken wings~~ chickenwings
Enter quantity : 200
Enter price per item: 150

The total party expense: 32400
```

2) Create a structure with name student with structure members : name , usn , gradelist of sem1 and gradelist of sem 2 . The student will be promoted to 3rd semester if he/she is not having backlog of credit count >= 16

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

    struct student {
            char name[50];
            int usn ;
             char S1grade[8];
             char S2grade[8];
       };


    int main() {
         int i, bck.cr = 0;
          struct student S1;
     printf(" Enter your name: ");
     scanf(" %s ", S1.name);
      printf(" USN: ");
     scanf(" %d ", S1.usn);
      printf(" SEM-1 grades in order : \n
        MCS-1 \n ACE \n IEE \n RES \n
        CAED \n   CEN\n   SFH \n   INC \n");
```

```
for ( i=0 ; i<8 ; i=i+1) {

        scanf ( "%c \n" , & ~~S1.grade[i]~~ S1.S1grade[i] );
  }


printf (" Sem 2 grades in order:
        MCS-2 \n  APS \n  ICE \n  POP \n
IWP \n  PWE \n  IDT \n  BLK ");


    ~~for ( int~~
    for( i=0 ; i<8 ; i=i+1) {
        scanf ("%c \n" , & S1.S2grade[i]);
    }


    for( i=0 ; i<8 ; i=i+1) {

            if ( i<2) {
    ~~printf ("S1.S1grade[i] == "F");~~

          if ( S1.S1grade[i] == "F" ) {
              bckcr = bckcr +4 ;
            }
          }
          if (i >=2 && i<5)
            {
              if( S1.S1 grade[i] == "F" )
                { bckcr = bckcr +3 ;
                }
            }
          if (i >=5 && i<8)
            {
```

```
if ( S1.S1grade[i] = = "F" )
      { bcker = bcker + 1 ;
      }
    }
}


for( i=0; i<8; i=i+1){
        if (i<2)
          { if ( S1.S2grade [i] = = "F" ){
                 bcker= bcker + 4 ;
                }
          }
        if (i>=2 && i<5) {
              if ( S1.S2grade[i] = = "F" ) {
                 bcker = bcker + 3;
                }
           }
          if ( i>=5 && i<8) {
               if ( S1.S2 grade[i] = = "F" ){
                   bcker = bcker+1 ;
                   }
                }
          }


  if ( bcker >= 16) {
         printf ("Not Eligible");
    }
   ~~else~~ else {
             printf(" Eligible");
      }
  return 0;  }
```

output

Enter your name : Adarsh

~~Name~~ USN : 1126934

SEM-1 grades in order:

MCS-1 F
ACS A+
IEE F
RES B
CAED F
CEN O
SFH B
INC C

SEM-2 grades in order:

MCS-2 F
APS B
ICE F
POP F
~~PWF~~ IWP B
~~IDT~~ PWE C
~~BLK~~ IDT A+
BLK O

Not Eligible

18|12|23

4)

->

```
for(i=0 ; i<a; i=i+1) {
        feetsum = feetsum +  dist.arr[i][0];
        inchsum = inchsum + dist.arr[i][1];
    }


printf(" total feet sum = %f \n", feetsum);
printf(" total inch sum = %f \n", inchsum);

return 0;
}
```

Output

```
Enter the number of inch-feet pairs to be
  added:  3
Enter the pairs:
  1
 2.5
  3.3
  3.1
 5.1
  2

total feet sum = 9.9
total inch sum = 7.6
```

3) Given an array arr[] containing N distance of the inch-feet system such that each element of the array represents a distance in form of {inch, feet}.
The task is to add all the N inch-feet distances using structures.

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

  struct distances{
          float arr[50][50];
      } dist;

  int main (){

    int a;
    float feetsum=0, inchsum=0;
    int i,j;
    printf(" Enter the number of inch-feet pairs to
              be added: ");
    scanf(" %d ", &a);

    printf(" Enter the pairs: \n");

      for ( i=0;i<a ; i=i+1){
          for( j=0 ; j<2 ;j=j+1){
      scanf(" %f ", & dist.arr[i][j];
          }
      }
```