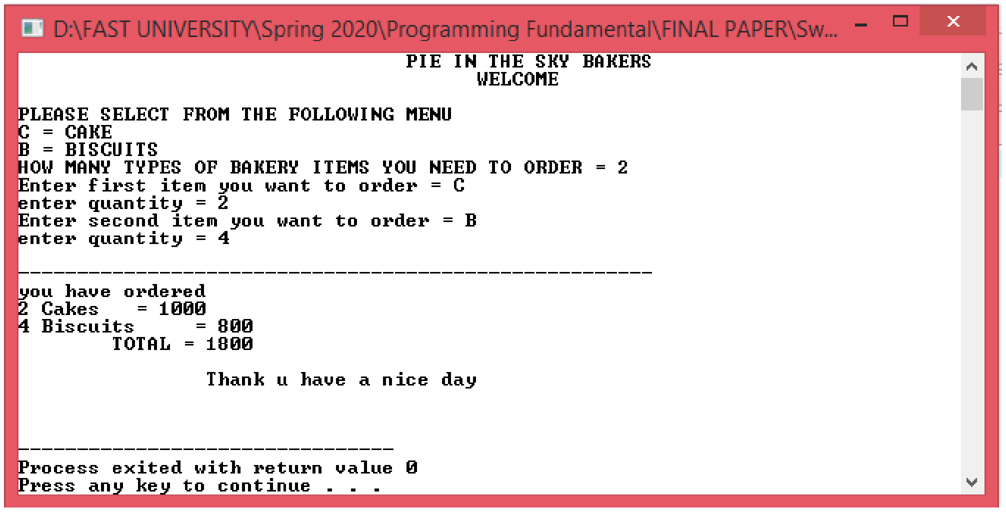
QUESTION:1

Welcome to the **"Pie in the Sky"** bakers. The customer wants to buy some combination of bakery items like any single item, or two items for example; cake and biscuits. Write a program that displays the following menu for the bakery items available to take order from the customer using IF and Switch case statement.

C = Cake	(Rs. 500)
B= Biscuits	(Rs. 200)

The program first asks to enter the no of categories of bakery items i.e. 1 or 2, then it asks to enter the choice i.e. 'C' for Cake, 'B' for Biscuits and then for quantity. If the customer asks for other than menu items, it shows message of the unavailability of item on bakers. The program should finally display the total charges for the order.

Output:

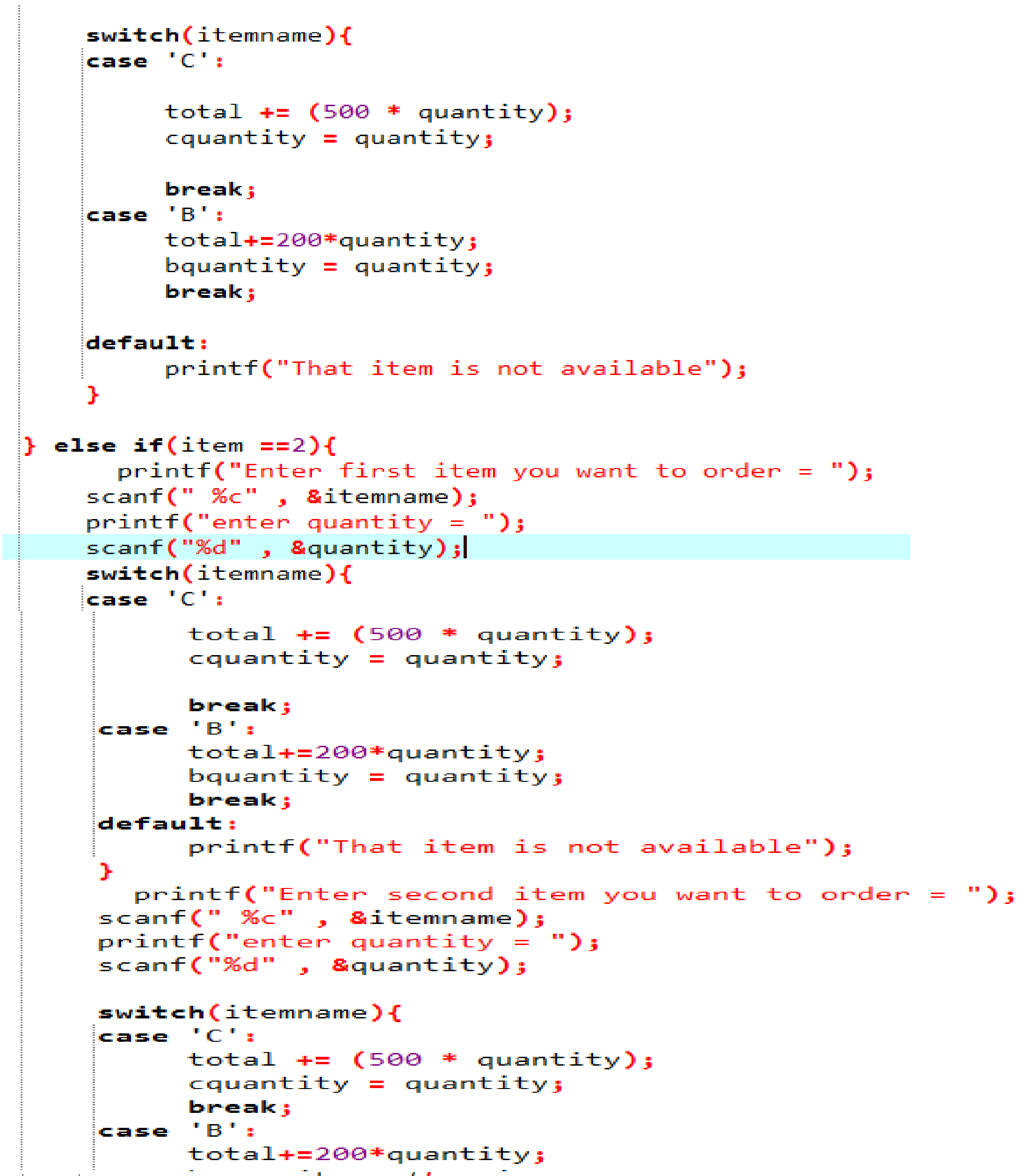


D:\FAST UNIVERSITY\Spring 2020\Programming Fundamental\FINAL PAPER\Sw...

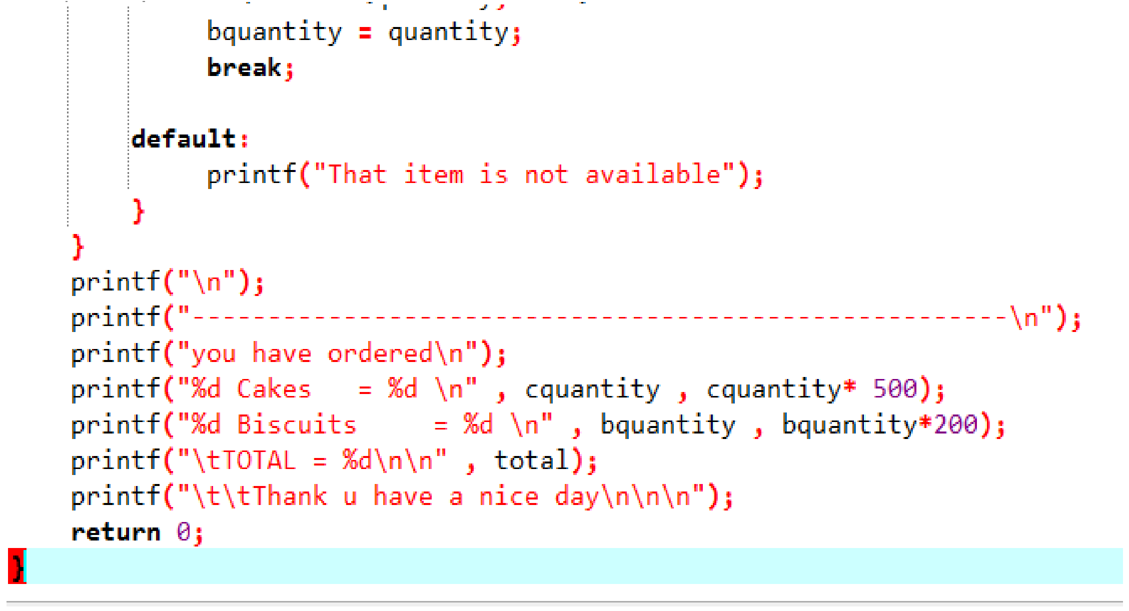
```
                                  PIE IN THE SKY BAKERS
                                        WELCOME

PLEASE SELECT FROM THE FOLLOWING MENU
C = CAKE
B = BISCUITS
HOW MANY TYPES OF BAKERY ITEMS YOU NEED TO ORDER = 2
Enter first item you want to order = C
enter quantity = 2
Enter second item you want to order = B
enter quantity = 4

------------------------------------------------------

you have ordered
2 Cakes    = 1000
4 Biscuits      = 800
        TOTAL = 1800

                Thank u have a nice day



--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Solution:

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    system("color F0");
    int total = 0;
    int item;
    char itemname;
    int cquantity=0;
    int bquantity=0;
    int quantity;
    printf("\t\t\t\t PIE IN THE SKY BAKERS\n");
    printf("\t\t\t\t       WELCOME\n\n");

    printf("PLEASE SELECT FROM THE FOLLOWING MENU\n");
    printf("C = CAKE\n");
    printf("B = BISCUITS\n");
    printf("HOW MANY TYPES OF BAKERY ITEMS YOU NEED TO ORDER = ");
    scanf("%d" , &item);
    if(item == 1){
        printf("Enter first item you want to order = ");
        scanf(" %c" , &itemname);
        printf("enter quantity = ");
        scanf("%d" , &quantity);
```



```c
    switch(itemname){
    case 'C':

        total += (500 * quantity);
        cquantity = quantity;

        break;
    case 'B':
        total+=200*quantity;
        bquantity = quantity;
        break;

    default:
        printf("That item is not available");
    }

} else if(item ==2){
      printf("Enter first item you want to order = ");
    scanf(" %c" , &itemname);
    printf("enter quantity = ");
    scanf("%d" , &quantity);
    switch(itemname){
    case 'C':

         total += (500 * quantity);
         cquantity = quantity;

         break;
     case 'B':
         total+=200*quantity;
         bquantity = quantity;
         break;
     default:
         printf("That item is not available");
     }
       printf("Enter second item you want to order = ");
     scanf(" %c" , &itemname);
     printf("enter quantity = ");
     scanf("%d" , &quantity);

     switch(itemname){
     case 'C':
         total += (500 * quantity);
         cquantity = quantity;
         break;
     case 'B':
         total+=200*quantity;
```

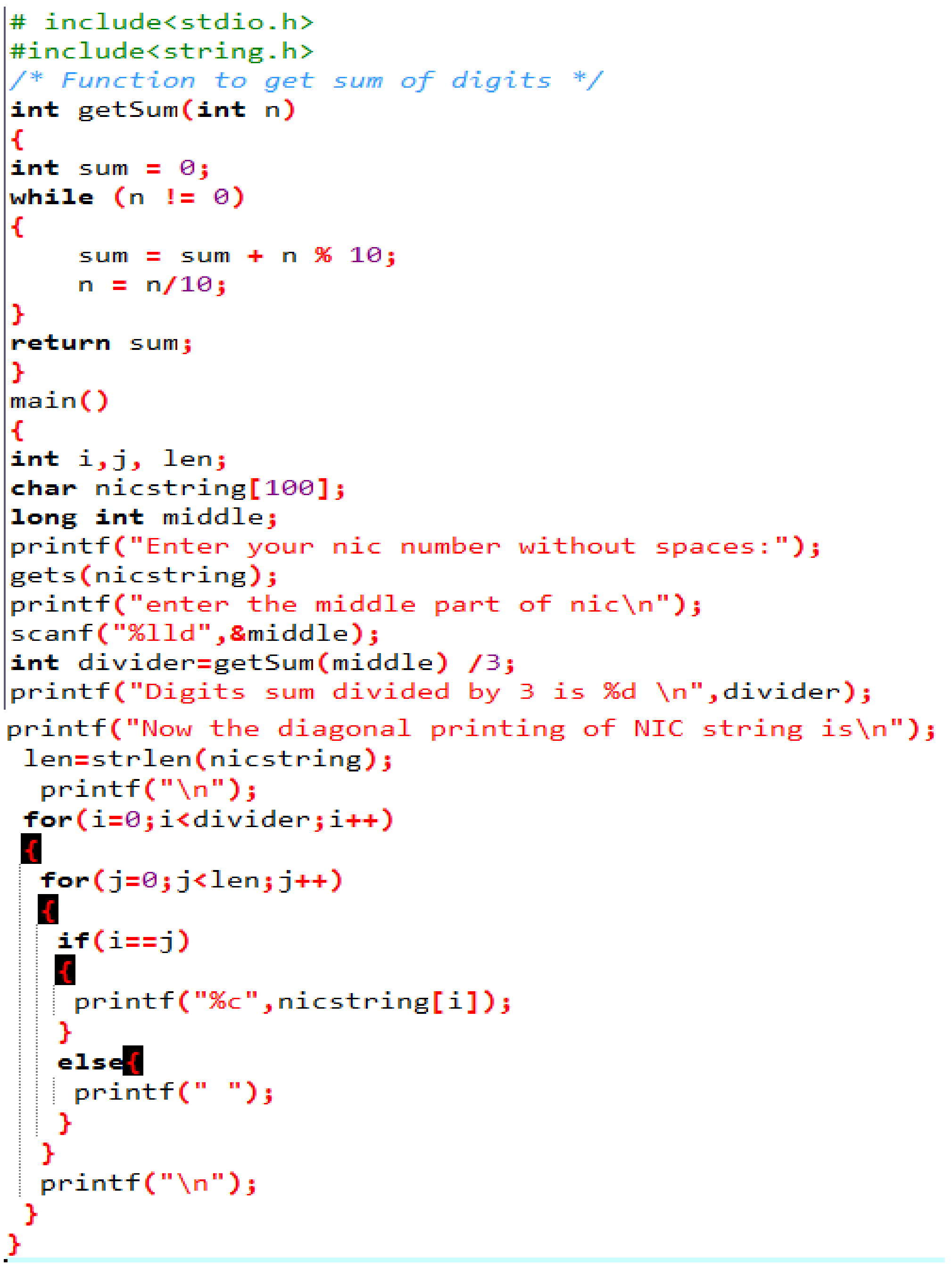


```c
        bquantity = quantity;
        break;

    default:
        printf("That item is not available");
    }
}
printf("\n");
printf("-----------------------------------------------------\n");
printf("you have ordered\n");
printf("%d Cakes   = %d \n" , cquantity , cquantity* 500);
printf("%d Biscuits     = %d \n" , bquantity , bquantity*200);
printf("\tTOTAL = %d\n\n" , total);
printf("\t\tThank u have a nice day\n\n\n");
return 0;
}
```

QUESTION: 2

Print your nic number string on the screen diagonally using nested loops by following the algorithm. Program should separate and add the individual digits of middle part of your NIC number ex. NIC number Input= “42213-1509923-0”. The program will input the whole NIC string 42213-1509923-0 and the middle number 1509923 and after separating the digits of the middle numbers add the digits to get the sum :1+5+0+9+9+2+3=29. Divide the sum by three using integer division which gives e.g. 29/3 =9. Now print your whole nic strings individual characters diagonally up to the final result i.e. 9 lines. Give the dry run of the whole algorithm:

Solution:



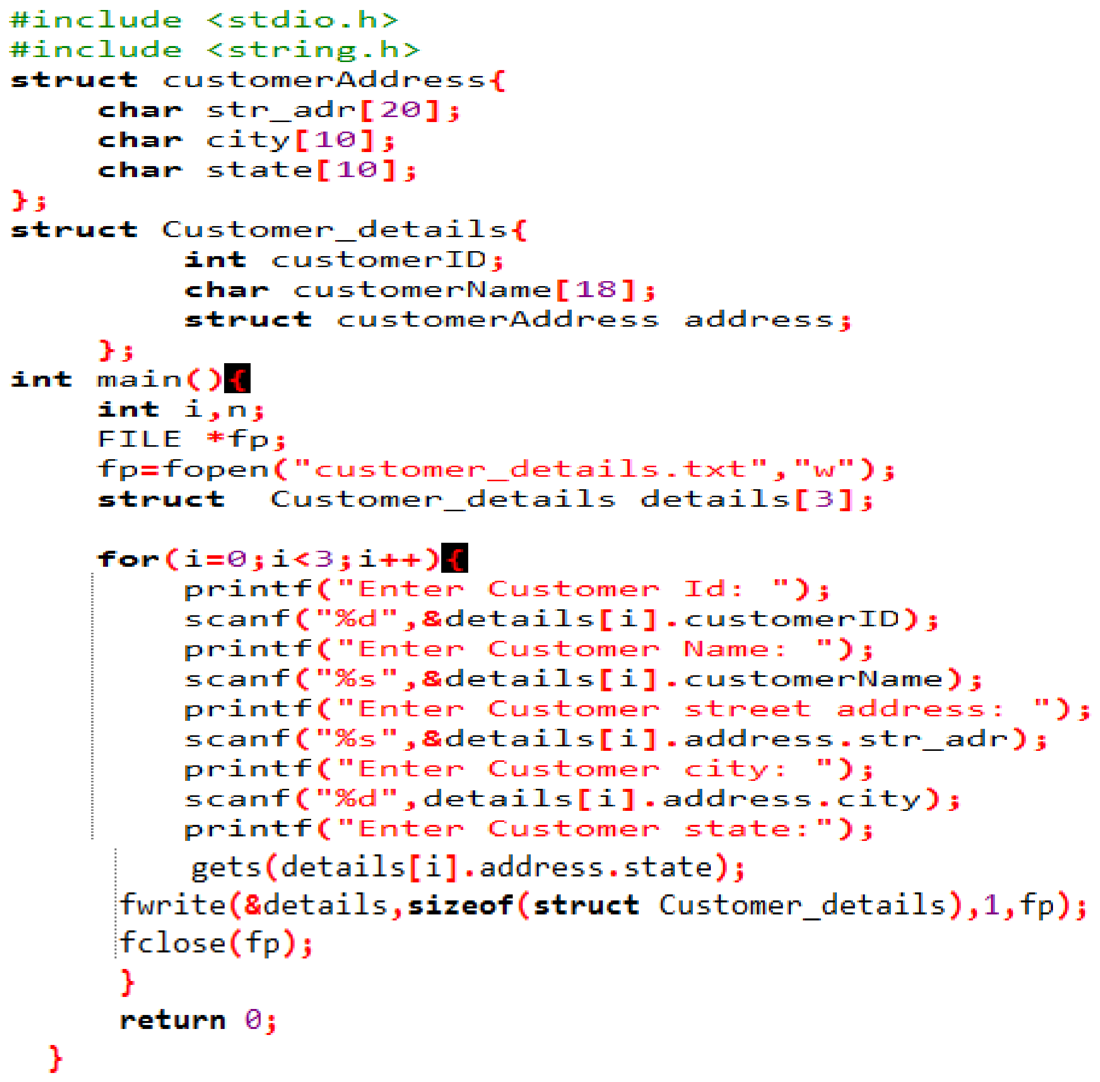
```c
# include<stdio.h>
#include<string.h>
/* Function to get sum of digits */
int getSum(int n)
{
int sum = 0;
while (n != 0)
{
    sum = sum + n % 10;
    n = n/10;
}
return sum;
}
main()
{
int i,j, len;
char nicstring[100];
long int middle;
printf("Enter your nic number without spaces:");
gets(nicstring);
printf("enter the middle part of nic\n");
scanf("%lld",&middle);
int divider=getSum(middle) /3;
printf("Digits sum divided by 3 is %d \n",divider);
printf("Now the diagonal printing of NIC string is\n");
 len=strlen(nicstring);
  printf("\n");
 for(i=0;i<divider;i++)
 {
  for(j=0;j<len;j++)
  {
   if(i==j)
   {
    printf("%c",nicstring[i]);
   }
   else{
    printf(" ");
   }
  }
  printf("\n");
 }
}
```

QUESTION: 3

A Super Market wants to maintain the record of customer visits, items available and quantity of items. The owner asks a software house for designing a software which easily enables its worker to input the record and list the record based on need. Suppose you are a software designer of that software house and you are assigned to do this task using structure and filing. Following are the requirements of the supermarket's owner which he wants.

a) The worker has to maintain the record of 'n' customers in (customer_details.txt) file by creating an array of structures called customer details which include data members— a customer ID (type int), a customer Name (type string), Customer address (consisting of street address, city, state). Create another structure to hold the customer address and include it as a nested structure. [The user has to input all the data members].

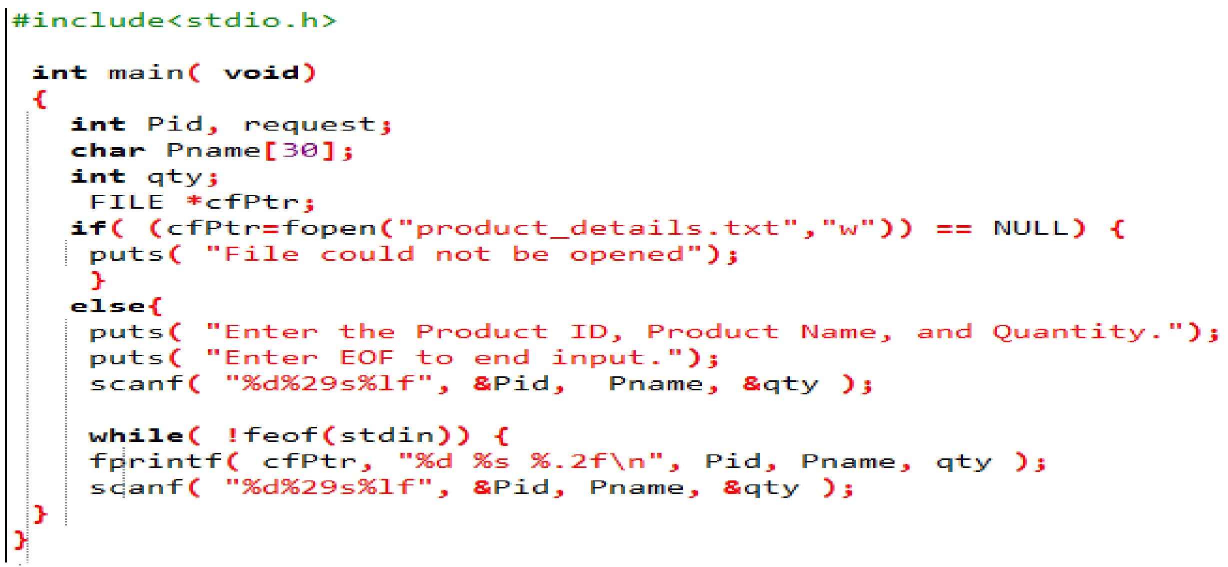
Solution:



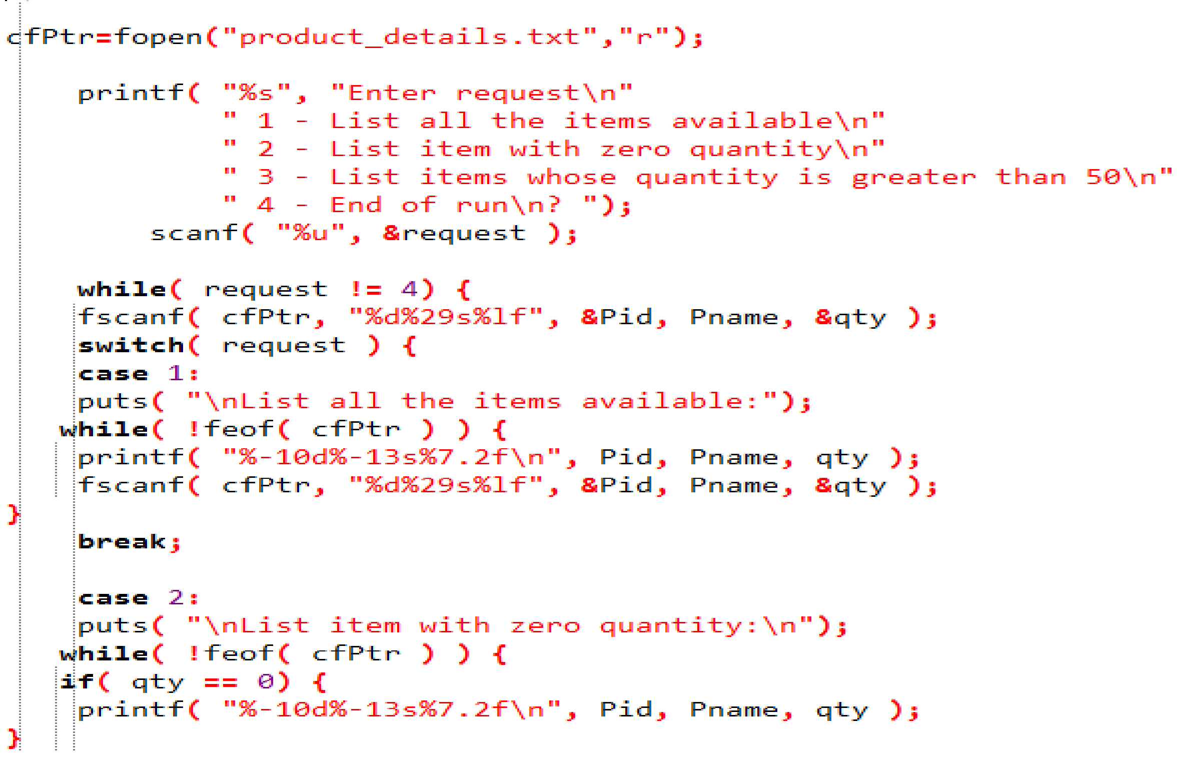
```
#include <stdio.h>
#include <string.h>
struct customerAddress{
    char str_adr[20];
    char city[10];
    char state[10];
};
struct Customer_details{
        int customerID;
        char customerName[18];
        struct customerAddress address;
    };
int main(){
    int i,n;
    FILE *fp;
    fp=fopen("customer_details.txt","w");
    struct  Customer_details details[3];

    for(i=0;i<3;i++){
        printf("Enter Customer Id:  ");
        scanf("%d",&details[i].customerID);
        printf("Enter Customer Name:  ");
        scanf("%s",&details[i].customerName);
        printf("Enter Customer street address:  ");
        scanf("%s",&details[i].address.str_adr);
        printf("Enter Customer city:  ");
        scanf("%d",details[i].address.city);
        printf("Enter Customer state:");
         gets(details[i].address.state);
     fwrite(&details,sizeof(struct Customer_details),1,fp);
     fclose(fp);
     }
     return 0;
  }
```

b) The worker has to maintain ID of Product, Name of Product, Quantity of Product in (Product_details.txt) file. He should be able to input the record until he doesn't press any special key for the end of input.
Further, the worker should have menu to select any option
i) The worker can list out all the maintained items at all once.
ii) The worker can list only those items whose quantity is zero.
iii) The worker can list only those items whose quantity is greater than fifty.

Solution:



```
#include<stdio.h>

 int main( void)
 {
    int Pid, request;
    char Pname[30];
    int qty;
     FILE *cfPtr;
    if( (cfPtr=fopen("product_details.txt","w")) == NULL) {
     puts( "File could not be opened");
     }
    else{
     puts( "Enter the Product ID, Product Name, and Quantity.");
     puts( "Enter EOF to end input.");
     scanf( "%d%29s%lf", &Pid,  Pname, &qty );

     while( !feof(stdin)) {
     fprintf( cfPtr, "%d %s %.2f\n", Pid, Pname, qty );
     scanf( "%d%29s%lf", &Pid, Pname, &qty );
 }
}
```



```
cfPtr=fopen("product_details.txt","r");

    printf( "%s", "Enter request\n"
               " 1 - List all the items available\n"
               " 2 - List item with zero quantity\n"
               " 3 - List items whose quantity is greater than 50\n"
               " 4 - End of run\n? ");
        scanf( "%u", &request );

    while( request != 4) {
    fscanf( cfPtr, "%d%29s%lf", &Pid, Pname, &qty );
    switch( request ) {
    case 1:
    puts( "\nList all the items available:");
   while( !feof( cfPtr ) ) {
    printf( "%-10d%-13s%7.2f\n", Pid, Pname, qty );
    fscanf( cfPtr, "%d%29s%lf", &Pid, Pname, &qty );
}
    break;

    case 2:
    puts( "\nList item with zero quantity:\n");
   while( !feof( cfPtr ) ) {
   if( qty == 0) {
    printf( "%-10d%-13s%7.2f\n", Pid, Pname, qty );
}
```

```
      fscanf( cfPtr, "%d%29s%lf", &Pid, Pname, &qty );
}
      break;

      case 3:
      puts( "\nList items whose quantity is greater than 50:\n");
    while( !feof( cfPtr ) ) {
    if( qty > 50){
          printf( "%-10d%-13s%7.2f\n", Pid, Pname, qty);
}
      fscanf( cfPtr, "%d%29s%lf", &Pid, Pname, &qty );
}
      break;
}// end switch
      rewind( cfPtr );
      printf( "%s", "\n? ");
      scanf( "%d", &request );
} // end while
      puts( "End of run");
      fclose( cfPtr );
}// end main
```

Question # 04:

During Covid-19 spread, civil hospital has to manage budget for its 6 departments. Its account manager has to track expense report : department-wise and month-wise from January to May in the following format:

Expense report

===

Department ----------------	January =======	February ========	March ========	April =======	May =======	**Total** ======
Surgery	27000.50	34000.50	45000.50	50000.50	50000.00	240002
Emergency	29000.50	27000.55	36000.00	36000.50	37000.50	201002.6
Skin	27000.50	36000.00	27000.00	36000.50	22000.50	170002
Covid-19	36000.50	36000.50	39000.50	27000.00	22000.00	182002
Dentistry	27000.50	36000.55	36000.00	22000.00	27000.00	170001.1
Total	146002.5	169002.1	183001	171001.5	158001	

Write a program using 2-dimensional arrays to hold the expense month-wise and department-wise.

a. Department and Months names should be stored and displayed using array of string pointers.
[Skipped]

b. Modular programming should be used i.e. input-data(parameters list..),output data(parameters list…), department-wise-total(parameters list…),month-wise-total(parameters list…) should be used to generate report. **[only two functions were said to design]**

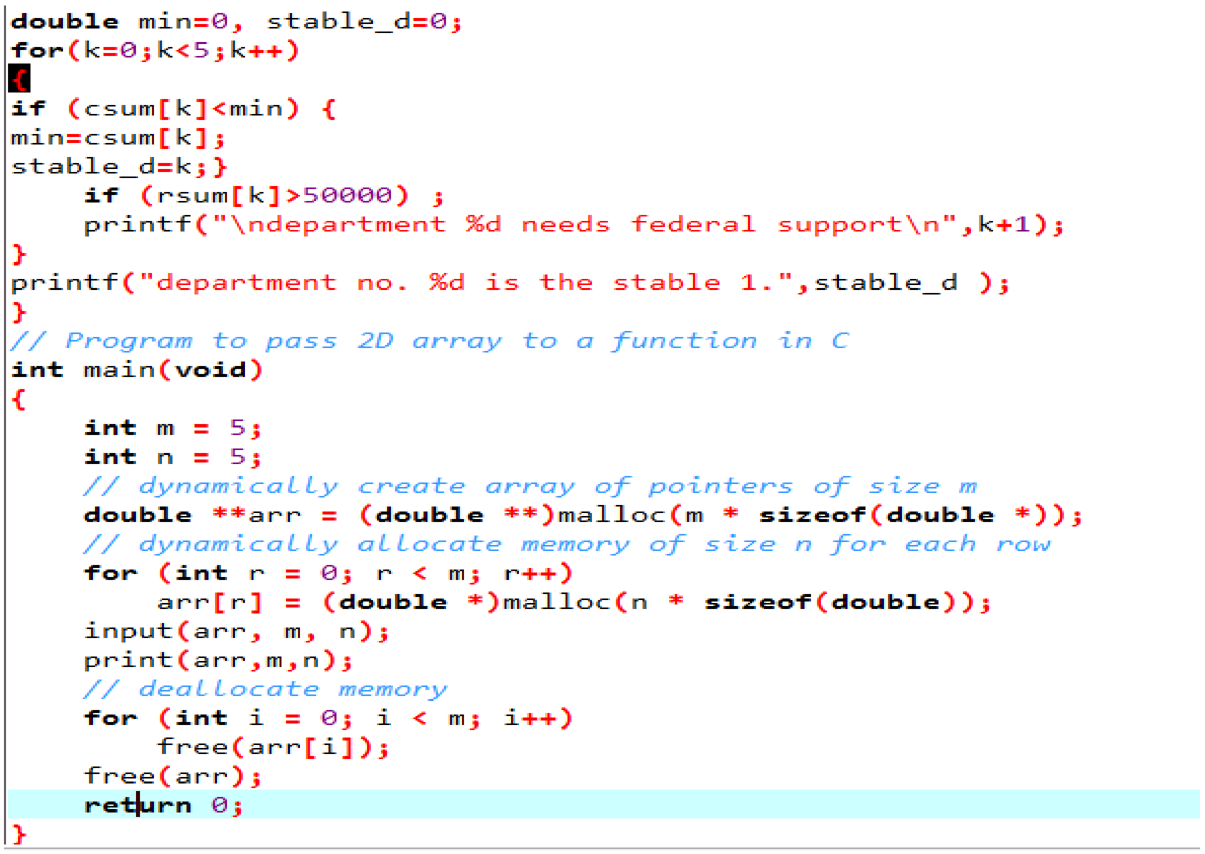
c. 2D array holding expenses should be passed from main to all functions by reference.

d. Calculate and display based on data, which department will need federal support i.e. expenses higher than 50,000.

e. Display which month was stable in terms of expenses i.e having the lowest monthly expense

Solution:

```
#include <stdio.h>
#include <stdlib.h>
// Here the parameter is an array of pointers
void input(double** arr, int m, int n)
{       double a[5][5]={{27000.50,34000.50,45000.50,50000.50,50000.00},
     {29000.50,27000.55,36000.00,36000.50,37000.50},
     {27000.50,36000.00,27000.00,36000.50,22000.50},
     {36000.50,36000.50,39000.50,27000.00,22000.00},
     {27000.50,36000.55,36000.00,22000.00,27000.00}};

        for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            arr[i][j] =a[i][j];
        }
    }
}

void print(double** arr, int m, int n){
// print 2D array
double rsum[5]={0} ,csum[5]={0};
int k;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            rsum[i] += arr[i][j]; k=j;
        }
        printf("\n");
    }
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < n; j++) {
            printf("%3.1f  ", arr[i][j]);
            csum[i] += arr[j][i];
        }
        printf("\n");
    }


printf("\t\t\t\t\t\tExpense Report\n");
printf("\t\t\t\t\t\t---------------\t\t\t\t\n");
printf("-----------January  February  March  April    May    Total\t\n");
printf("------------------=======  ========  =====  =====    ===    ======\t\n");
printf("Department-------------------------------------------------\t\n");
for (int c = 0;c < m; c++)
    { printf("----------");
    for (int d = 0; d < n+1; d++)
        {
        if(d<n) printf("%.1lf  ", arr[c][d]);
        else if (d==n)  printf("%.2lf\n",rsum[c]);
        }
    }
k=0;
printf("----------");
for( k=0;k<5;k++) {
printf("%.1f  ",csum[k]);}
```



```
double min=0, stable_d=0;
for(k=0;k<5;k++)
{
if (csum[k]<min) {
min=csum[k];
stable_d=k;}
    if (rsum[k]>50000) ;
    printf("\ndepartment %d needs federal support\n",k+1);
}
printf("department no. %d is the stable 1.",stable_d );
}
// Program to pass 2D array to a function in C
int main(void)
{
    int m = 5;
    int n = 5;
    // dynamically create array of pointers of size m
    double **arr = (double **)malloc(m * sizeof(double *));
    // dynamically allocate memory of size n for each row
    for (int r = 0; r < m; r++)
        arr[r] = (double *)malloc(n * sizeof(double));
    input(arr, m, n);
    print(arr,m,n);
    // deallocate memory
    for (int i = 0; i < m; i++)
        free(arr[i]);
    free(arr);
    return 0;
}
```

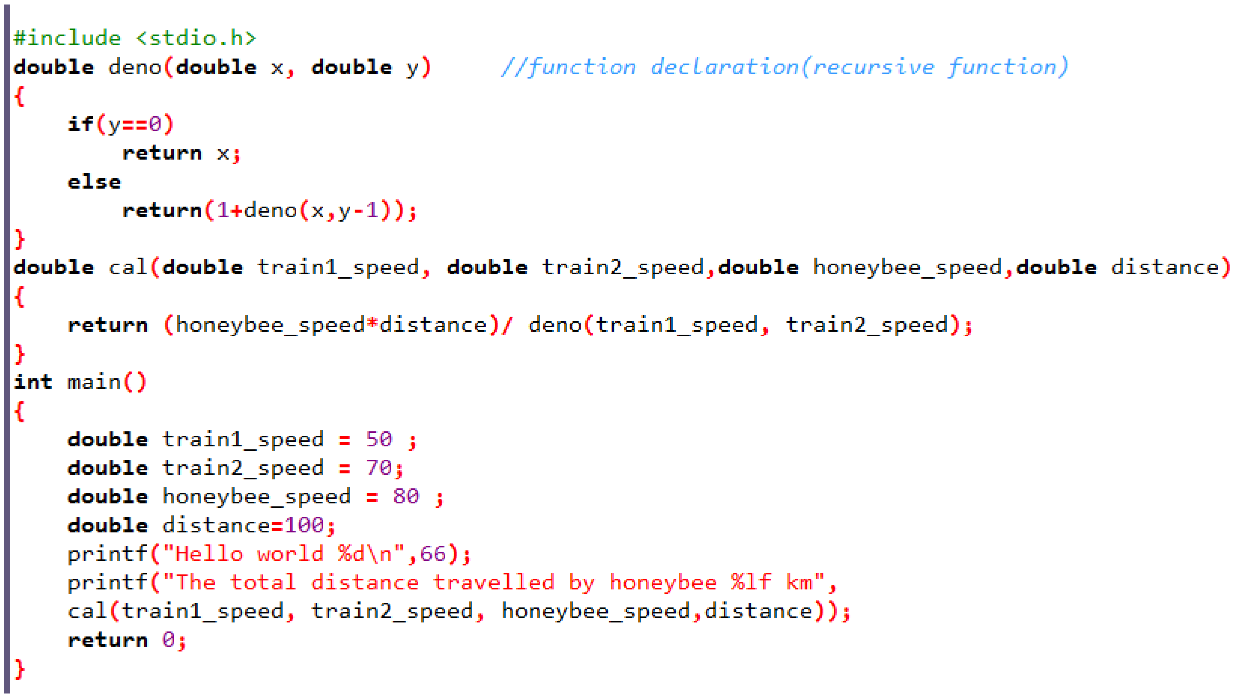
Question # 05:

Consider: **Two trains** are on the same track and they are coming toward each other. The speed of the first train is **50 km/h** and the speed of the second train is **70 km/h**. A honeybee starts flying between the trains when the distance between two trains is **100 km**. The honeybee first flies from first train to second train. Once it reaches the second train, it immediately flies back to the first train … and so on until **trains collide**. Calculate the total distance travelled by the honeybee. Speed of a honeybee is **80 km/h**. Distance formula is given below.

Total distance travelled by honeybee:
(honeybee speed * distance)/(train_1 speed + train_2 speed)

Your task is to solve the problem **recursively (direct or indirect)**.

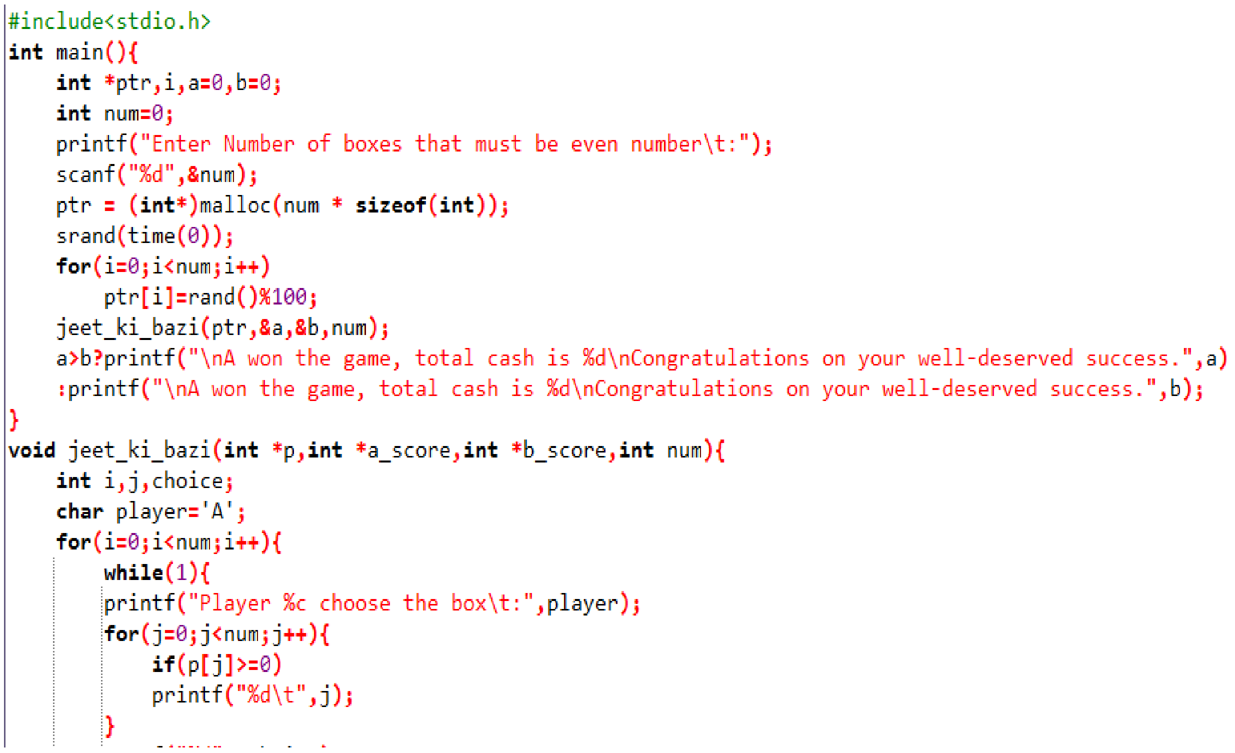
Solution:



```c
#include <stdio.h>
double deno(double x, double y)      //function declaration(recursive function)
{
    if(y==0)
        return x;
    else
        return(1+deno(x,y-1));
}
double cal(double train1_speed, double train2_speed,double honeybee_speed,double distance)
{
    return (honeybee_speed*distance)/ deno(train1_speed, train2_speed);
}
int main()
{
    double train1_speed = 50 ;
    double train2_speed = 70;
    double honeybee_speed = 80 ;
    double distance=100;
    printf("Hello world %d\n",66);
    printf("The total distance travelled by honeybee %lf km",
    cal(train1_speed, train2_speed, honeybee_speed,distance));
    return 0;
}
```

Question # 06:

Solutions:



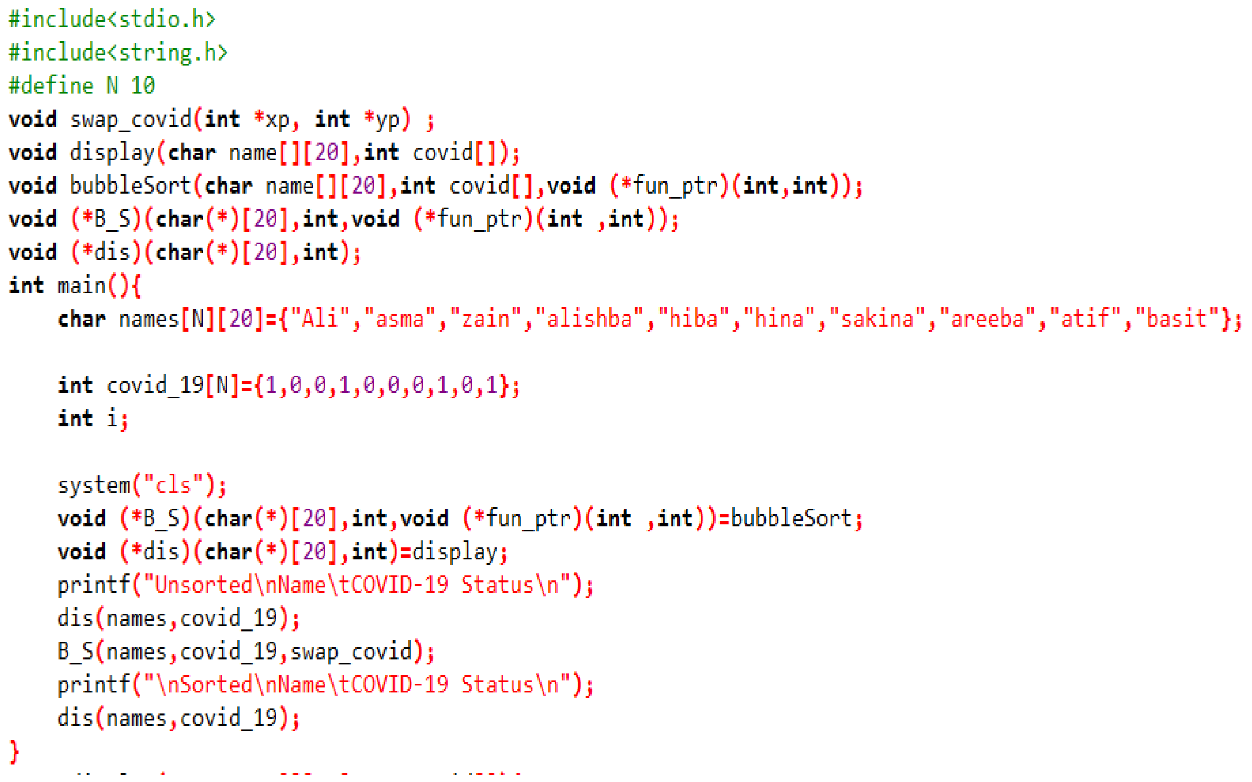
```c
#include<stdio.h>
int main(){
    int *ptr,i,a=0,b=0;
    int num=0;
    printf("Enter Number of boxes that must be even number\t:");
    scanf("%d",&num);
    ptr = (int*)malloc(num * sizeof(int));
    srand(time(0));
    for(i=0;i<num;i++)
        ptr[i]=rand()%100;
    jeet_ki_bazi(ptr,&a,&b,num);
    a>b?printf("\nA won the game, total cash is %d\nCongratulations on your well-deserved success.",a)
    :printf("\nA won the game, total cash is %d\nCongratulations on your well-deserved success.",b);
}
void jeet_ki_bazi(int *p,int *a_score,int *b_score,int num){
    int i,j,choice;
    char player='A';
    for(i=0;i<num;i++){
        while(1){
        printf("Player %c choose the box\t:",player);
        for(j=0;j<num;j++){
            if(p[j]>=0)
            printf("%d\t",j);
        }
        }
        scanf("%d",&choice);
        if(p[choice]>=0 && player=='A'){
            *a_score+=p[choice];
            p[choice]=-1;
            player='B';
            break;
        }
        else if(p[choice]>=0 && player=='B'){
            *b_score+=p[choice];
            p[choice]=-1;
            player='A';
            break;
        }
        else{
            printf("already opened. Choose from remaing boxes\n");
            continue;
        }
    }
    }
}
```

Question # 07:

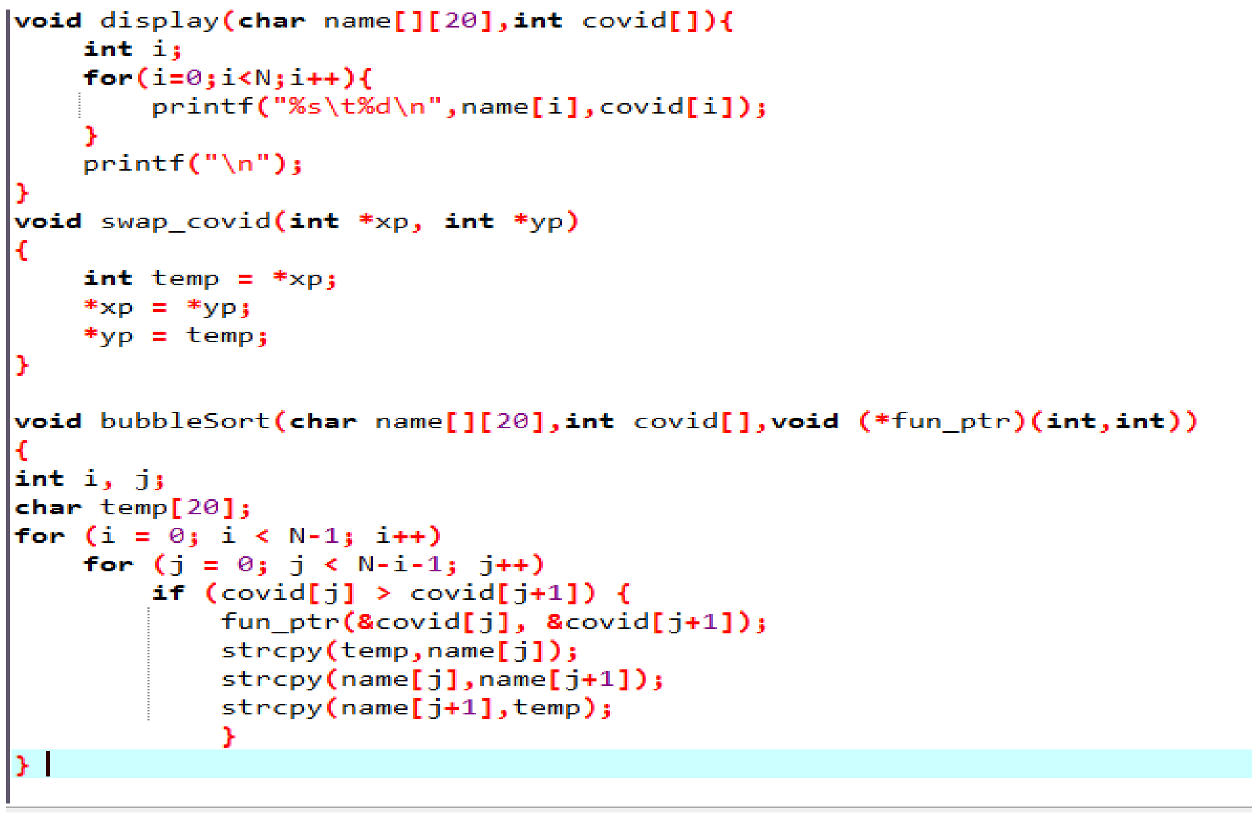
Solution:

```
#include <string.h>
#include <stdio.h>
int main () {
   char str[100];
   //= "This is Maria stupid - Full of ugly and pathetic ideas . Alas , she is here to give us company!";
   const char s[2] = " ";
   char *token;
  printf("Enter an input string: \n") ;
  gets(str);
   /* get the first token */
   token = strtok(str, s);
   printf("\nOutput string: ");
   /* walk through other tokens */
   while( token != NULL ) {
      if(!strcmp(token,"stupid"))
      printf( " %s ", "the great" );
      else if(!strcmp(token,"ugly"))
      printf( " %s\ ", "vibrant" );
      else if(!strcmp(token,"pathetic"))
      printf( " %s ", "shiny" );
      else if(!strcmp(token,"Alas"))
      printf( " %s ", "Wow" );
      else printf("%s ",token);

      token = strtok(NULL, s);
   }
   return(0);
}
```

Question # 08: **Solution**



```c
#include<stdio.h>
#include<string.h>
#define N 10
void swap_covid(int *xp, int *yp) ;
void display(char name[][20],int covid[]);
void bubbleSort(char name[][20],int covid[],void (*fun_ptr)(int,int));
void (*B_S)(char(*)[20],int,void (*fun_ptr)(int ,int));
void (*dis)(char(*)[20],int);
int main(){
    char names[N][20]={"Ali","asma","zain","alishba","hiba","hina","sakina","areeba","atif","basit"};

    int covid_19[N]={1,0,0,1,0,0,0,1,0,1};
    int i;

    system("cls");
    void (*B_S)(char(*)[20],int,void (*fun_ptr)(int ,int))=bubbleSort;
    void (*dis)(char(*)[20],int)=display;
    printf("Unsorted\nName\tCOVID-19 Status\n");
    dis(names,covid_19);
    B_S(names,covid_19,swap_covid);
    printf("\nSorted\nName\tCOVID-19 Status\n");
    dis(names,covid_19);
}
```



```c
void display(char name[][20],int covid[]){
    int i;
    for(i=0;i<N;i++){
        printf("%s\t%d\n",name[i],covid[i]);
    }
    printf("\n");
}
void swap_covid(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void bubbleSort(char name[][20],int covid[],void (*fun_ptr)(int,int))
{
int i, j;
char temp[20];
for (i = 0; i < N-1; i++)
    for (j = 0; j < N-i-1; j++)
        if (covid[j] > covid[j+1]) {
            fun_ptr(&covid[j], &covid[j+1]);
            strcpy(temp,name[j]);
            strcpy(name[j],name[j+1]);
            strcpy(name[j+1],temp);
            }
}
```