

(1) Write a program to print addition of two 3×3 matrix

(a) #include <stdio.h>
#include <conio.h>
Void main () {
int a[3][3], b[3][3], i, j;

Printf ("1+1) + Enter the value of first matrix\n\n");

for(i=0; i<3; i++) {
 for(j=0; j<3; j++) {
 Printf ("\n Enter the value of row%d and
 Column %d: ", i+1, j+1);
 Scany ("%d", &a[i][j]);
 }
}

Clscr();

Printf ("1+1) + Enter the value of Second matrix\n\n");

for(i=0; i<3; i++) {
 for(j=0; j<3; j++) {
 Printf ("\n Enter the value of row%d and
 Column %d: ", i+1, j+1);
 Scany ("%d", &b[i][j]);
 }
}

System.Clsscr();

```
Printf("The value of first matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        Printf("%d\t", a[i][j]);
    }
    Printf("\n");
}
```

```
Printf("\n\nThe value of second matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        Printf("%d\t", b[i][j]);
    }
    Printf("\n");
}
```

```
Printf("\n\nThe addition of both matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        Printf("%d\t", a[i][j] + b[i][j]);
    }
    Printf("\n");
}
```

getch();

(2) Write a program to create 3×3 matrix of numeric value. Find inputted value by user in this matrix. If user's value isn't found, then print proper message.

3)

```
#include <Stdio.h>
```

```
#include <Conio.h>
```

```
Void main () {
```

```
    int a[3][3], i, j, n, flag = 0;
```

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

```
        for (j = 0; j < 3; j++) {
```

```
            printf ("\n Enter the value of a[%d][%d]:",
```

```
                    i + 1, j + 1);
```

```
            scanf ("%d", &a[i][j]);
```

```
}
```

```
    clrscr ();
```

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

```
        for (j = 0; j < 3; j++) {
```

```
            printf ("%d\t", a[i][j]);
```

```
}
```

```
    printf ("\n");
```

```
}
```

```
    printf ("\n\n\n\n\nEnter the element you want  
to search: ");
```

```
Sccanf ("%d", &n);
```

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

```
    for (j=0; j<3; j++) {
```

```
        if (a[i][j] == n) {
```

```
            printf ("\n%d is present at %d row,  
                    %d column of the matrix (%d,%d),  
                    %d, %d, %d, %d);
```

```
            flag = 1;
```

```
}
```

```
}
```

```
if (flag == 0) {
```

```
    printf ("\n\n%d is not present in the  
            given matrix!", n);
```

```
getch();
```

```
}
```

(3) Write a program to print multiplication of two 3×3 matrix.

3) #include <stdio.h>

include <Conio.h>

Void main () {

int a[3][3], b[3][3], result[3][3], sum=0, i, j, k;

Printf ("Enter the value of first matrix\n");

for (i=0; i<3; i++) {

 for (j=0; j<3; j++) {

 Printf ("\nEnter the value of row %d, Column %d:", i+j, j+1);

 Scansf ("%d", &a[i][j]);

}

}

Close();

Printf ("\nEnter the value of second matrix");

for (i=0; i<3; i++) {

 for (j=0; j<3; j++) {

 Printf ("\nEnter the value of row %d, Column %d:", i+1, j+1);

 Scansf ("%d", &b[i][j]);

}

}

```

for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        for (k=0; k<3; k++) {
            sum += a[i][k] * b[k][j];
            result[i][j] = sum;
        }
    }
}
Sum = 0;
}
Close();

```

```

printf("The Value of first matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        printf("%d\t", a[i][j]);
    }
    printf("\n");
}
printf("\n\n");
printf("Value of second matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        printf("%d\t", b[i][j]);
    }
    printf("\n");
}

```

```

printf("\n\nThe multiplication of first and
second matrix is:\n");
for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
        printf("%d\t", result[i][j]);
    }
}

```

```
    }  
    printf("\n");  
}  
getch();
```

(4) Write a program to enter 5 student name and search inputted user in it.

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

```
void main () {
```

```
    Char [5][50], input[50];
    int i, flag=0;
```

```
    printf ("\t\t\tName of Students\n\n");
```

```
    for (i=0; i<5; i++) {
```

```
        printf ("\nEnter name of Student :d:", i+1);
```

```
        gets (S[i]);
```

```
    clrscr();
```

```
    for (i=0; i<5; i++) {
```

```
        printf ("\n%d. %s", i+1, S[i]);
```

```
    printf ("\n\n\n\tEnter the name of the student  
you want to search: ");
```

```
    gets (input);
```

```
    for (i=0; i<5; i++)
```

```
if (Strncmp(input, s[i]) == 0) {
    printf ("\n\n%s is present at Place
            %d", input, i+1);
    flag = 1;
}
if (flag == 0) {
    printf ("\n\nName not found in the
            records!");
}
getch();
```

(5) Write a program to enter 5 student names and Print it in descending order.

```
#include <stdio.h>
#include <Conio.h>
#include <String.h>
void main(){
    char S[5][50], t[50];
    int i, j;
    for (i=0; i<5; i++){
        printf("\nEnter string %d", i+1);
        gets(S[i]);
    }
    for (i=0; i<5; i++){
        for (j = i+1; j<5; j++){
            if (strcmp(S[i], S[j])>0){
                strcpy(t, S[i]);
                strcpy(S[i], S[j]);
                strcpy(S[j], t);
            }
        }
    }
    printf("Output string (Sorted)");
    for (i=0; i<5; i++){
        printf("\n%s", S[i]);
    }
    getch();
}
```

(6) Write a program to Create Structure marksheet (Sid, Sname, Sub_1, Sub_2, Sub_3, total, Per). Print Only one Student marksheet.

s6) #include <stdio.h>
#include <string.h>
#include <conio.h>

```
struct marksheet {  
    int S_id ;  
    char S_name[50];  
    float Sub_1 ; Sub_2 , Sub_3 , total , Percentage ;  
};
```

```
void main (){  
    struct marksheet s1 ;  
    printf ("\t\t\tEnter student detail");
```

```
    printf ("\n\n Enter ID: ");  
    scanf ("%d", &s1.S_id);
```

```
    getch();
```

```
    printf ("\n\n Enter name: ");  
    gets (s1.S_name);
```

```
    printf ("\n\n Enter Sub_1 marks: ");  
    scanf ("%f", &s1.Sub_1);
```

```
    printf ("\n\n Enter Sub_2 marks: ");
```

```
Scansf ("%f", &S1.Sub_2);
```

```
Printf ("\n\n Enter Sub_3 marks: ");  
Scansf ("%f", &S1.Sub_3);
```

```
S1.total = S1.Sub_1 + S1.Sub_2 + S1.Sub_3;  
S1.percentage = (S1.total / 300) * 100;
```

```
Clrscr();
```

```
Printf ("\\t\\t\\tInformation of Student");
```

```
Printf ("\n\\nS_ID\\tS_name\\tSub_1\\tSub_2\\tSub_3  
\\ttotal\\tpercentage");
```

```
Printf ("\n\\n\\n%.d\\t%.s\\t%.0.1f\\t%.0.1f\\t%.0.1f\\t  
%.0.1f\\t%.0.2f", S1.S_ID, S1.S_name, S1.Sub_1,  
S1.Sub_2, S1.Sub_3, S1.total, S1.percentage);
```

```
getch();
```

```
}
```

(7) Write a program to Create Structure marksheets(S_id, S_name, Sub_1, Sub_2, Sub_3, total, per). Print 5 student marksheets.

7) #include <stdio.h>

#include <string.h>

#include <conio.h>

struct marksheets {

int S_id;

char name[50];

float Sub_1, Sub_2, Sub_3, total, Percentage;

};

int main()

Struct marksheets S[5];

int i;

for (i=0; i<5; i++) {

printf ("\\t\\t Enter details of student %d", i+1);

printf ("\n\n Enter the ID of the student: ");

scanf ("%d", &S[i].S_ID);

getch();

printf ("\n\n Enter name: ");

gets (S[i].S_name);

printf ("\n\n Enter Sub-1 marks: ");

```
Scansf ("%f", & S[i].Sub_1);
```

```
printf ("lnlnEnter Sub-2 marks: ");  
Scansf ("%f", & S[i].Sub_2);
```

```
printf ("lnlnEnter Sub-3 marks: ");  
Scansf ("%f", & S[i].Sub_3);
```

```
S[i].total = S[i].Sub_1 + S[i].Sub_2 + S[i].Sub_3;
```

```
S[i].percentage = (S[i].total / 300) * 100;
```

```
} class();
```

```
Printf ("lflfl Information of Student");
```

```
Printf ("lnlnS_IDlS_name l Sub-1 l Sub-2 l Sub-3 l  
total l Percentage");
```

```
for (i=0 ; i<5 ; i++) {
```

```
Printf ("lnln%.d l %.S l %.1f l %.1f l %.1f l %.1f  
l %.2f", S[i].S_id, S[i].S_name,  
S[i].Sub_1, S[i].Sub_2, S[i].Sub_3,  
S[i].total, S[i].percentage);
```

```
} getch();
```

```
}
```

(8) Write a program to Create structure inventory (i_id, i_name, qty, rate, amount). Enter 5 items in inventory. Search item based on its i_id and print it, if available, otherwise print proper message.

(8)

```
#include <Stdio.h>
```

```
#include <String.h>
```

```
#include <Conio.h>
```

```
Struct inventory {
```

```
    int i_id, qty, rate, amount;
```

```
    Char i_name[50];
```

```
};
```

```
Void main (){
```

```
    Struct inventory inv[5];
```

```
    int i, n, flag=0;
```

```
    for(i=0; i<5; i++){
```

```
        printf("Enter info of inventory %d\n",  
               i+1);
```

```
        printf("Enter ID: ");
```

```
        scanf("%d", &inv[i].i_id);
```

```
        getch();
```

```
        printf("Enter name: ");
```

```
        gets(inv[i].i_name);
```

```
    printf ("\nEnter quantity: ");
    scanf ("%d", &inv[i].qty);
```

```
    printf ("\nEnter rate per item: ");
    scanf ("%d", &inv[i].rate);
```

```
    inv[i].amount = inv[i].qty * inv[i].rate;
```

```
} System ("cls");
```

```
printf ("ID\tName\tQty\tRate\tAmount\n");
for (i=0; i<5; i++) {
```

```
    printf ("%d\t%s\t%d\t%.2f\t%.2f\n", inv[i].  
            i_id, inv[i].i_name, inv[i].qty,  
            inv[i].rate, inv[i].amount);
}
```

```
printf ("Input ID to search: ");
scanf ("%d", &n);
```

```
printf ("\n\nID\tName\tQty\tRate\t  
        Amount\n");
```

```
for (i=0; i<5; i++) {
```

```
    if (i == n) {
        printf ("%d\t%s\t%d\t%.2f\t%.2f\n", inv[i].  
                i_id, inv[i].i_name, inv[i].qty,  
                inv[i].rate, inv[i].amount);
```

```
    } } flag = 1;  
}  
  
if (flag == 0){  
    printf ("\\n Entry not found in the  
    records!");  
}  
  
getch();  
}
```

(9) Write a program to Create structure inventory (i_id, name, qty, rate, amount). Enter 5 items in inventory. Search items based on it's name. if not found, print proper message.

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

struct inventory {
    int i_id, qty, rate, amount;
    Char i_name[50];
};

void main () {
    struct inventory inv[5];
    int i, n, flag = 0;
    Char input[50];

    for (i=0; i<5; i++) {
        printf ("Enter info of inventory %d\n", i+1);
        printf ("\nEnter ID: ");
        scanf ("%d", &inv[i].i_id);
        getchar ();
        printf ("\nEnter name: ");
        gets (inv[i].i_name);
        printf ("\nEnter quantity: ");
    }
}
```

```
scanf ("%d", &inv[i].qty);
```

```
printf ("\nEnter rate per item: ");  
scanf ("%d", &inv[i].rate);
```

```
inv[i].amount = inv[i].qty * inv[i].rate;
```

```
} clrscr();
```

```
printf ("I-ID\tI-name\tqty\tRate\tamount\n");
```

```
for (i=0 ; i<5 ; i++){
```

```
    printf ("%d\t%s\t%d\t%d\t%d\n", inv[i].I_ID,  
           inv[i].i_name, inv[i].qty, inv[i].rate,  
           inv[i].amount);
```

```
}
```

```
getch();
```

```
printf ("\n\n\tEnter the name you want  
to search: ");
```

```
gets (input);
```

```
printf ("\n\n\tI-ID\tI-name\tQty\tRate\tAmount  
\n");
```

```
for (i=0; i<5; i++){
```

```
    if (strcmp (inv[i].i_name, input) == 0){
```

```
        printf ("\n\t%d\t%s\t%d\t%d\t%d\n", inv[i]
```

```
            .i_id, inv[i].i_name, inv[i].qty)
```

```
iwm[i].rate, iwm[i].amount);  
    } flag = 1;  
}  
  
if (flag == 0){  
    printf ("InvEntry not found in the records!");  
}  
  
getch();  
}
```

(10) Write a menu driven program on structure inventory (i_id, i_name, qty, rate, amount). Enter 5 items in inventory. Perform the following operations :-

1. Search items where qty > 5
2. Search items where rate < 10 and item name starts with 'P'.
3. Sort items on item name (Ascending order).

Ans 10) #include <stdio.h>
#include <string.h>
#include <conio.h>

Struct inventory {
 int i_id, qty, rate, amount;
 char i_name[50];
};

void main(){
 Struct inventory inv[5];
 int i, j, n, temp1, flag=0;
 char temp[50];

 for (i=0; i<5; i++){
 printf("\t\t\tEnter info of inventory %d\n", i+1);
 printf("\n Enter ID: ");
 scanf("%d", &inv[i].i_id);
 getchar();
 printf("\n Enter name: ");

```
gets (inv[i].i_name);
```

```
Printf ("\nEnter quantity: ");  
Scanf ("%d", &inv[i].qty);
```

```
Printf ("\nEnter rate per item: ");  
Scanf ("%d", &inv[i].rate);
```

```
inv[i].amount = inv[i].qty * inv[i].rate;
```

```
} clrscr();
```

```
Printf ("I ID | I name | I Qty | I Rate | I amount |n");
```

```
for (i=0; i<5; i++) {
```

```
Printf ("|n|n% d | % s | % d | % d | % d |n", inv[i].  
i_id, inv[i].i_name, inv[i].qty, inv[i].  
rate, inv[i].amount);
```

```
}
```

```
Printf ("\n\nPlease choose your choice from the  
following :-\n");
```

```
Printf ("1. Search items where quantity is more  
than 5\n2. Search items where rate is  
less than 10 and item name starts with  
'P'\n3. Sort items in ascending order  
according to item name\n");
```

```
Printf ("\n\n\n\n1) Enter your choice: ");
```

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

```
switch (n) {
```

Case 1:

```
printf ("\n\n\n ID | name | Qty | Rate,\n | Amount |\n");
```

```
for (i=0; i<5; i++) {
```

```
if (inv[i].qty > 5) {
```

```
printf ("\n %d | %s | %d | %d | %d\n", inv[i].  
i-id, inv[i].i-name, inv[i].  
qty, inv[i].rate, inv[i].  
amount);
```

```
} flag = 1;
```

```
}
```

```
if (flag == 0) {
```

```
printf ("There is no item with quantity  
more than 5!");
```

```
}
```

```
break;
```

Case 2:

```
printf ("\n\n\n ID | name | Qty | Rate  
| amount |\n");
```

```
for (i=0; i<5; i++) {
```

```
if (inv[i].rate < 10 && inv[i].
```

{
 i-name[0] == 'P' || *inv*[*i*].*i-name*[0] == 'x')

printf ("%.d\t%.s\t%.d\t%.d\t%.d\n", *inv*[*i*].
 i-id, *inv*[*i*].*i-name*, *inv*[*i*].*qty*,
 inv[*i*].*rate*, *inv*[*i*].*amount*);

} flag = 1;
}

if (f == 0){

printf ("There is no item with rate less
 than 10 and whose name starts
 with P!");

}

break;

Case 3 :

for (*i* = 0; *i* < 5; *i*++) {

 for (*j* = *i* + 1; *j* < 5; *j*++) {

 if (*strcmp* (*inv*[*i*].*i-name*, *inv*[*j*].*i-name*)
 > 0) {

 temp1 = *inv*[*i*].*i-id*;

inv[*i*].*i-id* = *inv*[*j*].*i-id*;

inv[*j*].*i-id* = temp1;

strcpy (temp, *inv*[*i*].*i-name*);

strcpy (*inv*[*i*].*i-name*, *inv*[*j*].*i-name*);

strcpy (*inv*[*j*].*i-name*, temp);

 temp1 = *inv*[*i*].*qty*;

```
inv[i].qty = inv[j].qty;  
inv[j].qty = temp1;
```

```
temp1 = inv[i].rate;  
inv[i].rate = inv[j].rate;  
inv[j].rate = inv.temp1;
```

```
temp1 = inv[i].amount;  
inv[i].amount = inv[j].amount;  
inv[j].amount = temp1;
```

}
}

```
printf("I_ID\|I_name\|Qty\|Rate\| amount\n");
```

```
for (i=0 ; i<5 ; i++) {
```

```
    printf ("%d\|%s\|%.1f\|%.1f\|%.1f\n", inv[i].i_id,  
           inv[i].i_name, inv[i].qty, inv[i].rate,  
           inv[i].amount);
```

}

```
break;
```

```
default :
```

```
printf ("\n\nError! Please choose from one  
of the options above!");
```

}

```
getch();
```