Sankalp Mohanty
CSE-3
21053218

DSA(L)
6/8/22

28

Assignment-3

1. WAP to accept name, age and city of five people and display the name and city of eldest one.

Sol^n:

```
#include <stdio.h>

struct Person
{
  char name[100];
  int age;
  char city[100];
};

int main()
{
 struct Person p[5];
 int i;
printf("Enter the name, age, city name :- \n");
for (i=0; i<5; i++)
{
 scanf("%s", &p[i].name);
scanf("%d", &p[i].age);
scanf("%s", &p[i].city);
printf("\n");
}
struct Person elder;
int max = 0;
for (i=0; i<5; i++)
{
    if (p[i].age > max)
   {
    max = p[i].age;
    elder = p[i];
  }
```

```
printf("The name of the person, age(oldest one) and city
                                name is: %s %d %s",
                                elder.name, max, elder.city);
}
```

OUTPUT

```
Enter the name, age, city name :-
Arjun 34 Delhi
Dash 31 Mumbai
Ranga 42 Bhubaneswar
Bhishma 43 Hyderabad
Vijay 49 Chennai
The name of the person, age(oldest one) and city name is: Vijay 49 Chenna
```

2. WAP to accept name and arrival time of five trains and display the name with Railway time format.

Sol^n:-

```
#include <stdio.h>
# include <string.h>
struct timer
 {
 int val;
 int min;
 char pre[2];
 };
struct train
 {
  char name [100];
  struct timer tim;
 };
void getData (struct train t[5])
 {
 for (int i = 0; i <5; i++)
 {
  printf(" Enter the name of the train %d = ", i);
  scanf ("%s", &t[i].name);
  printf (" Enter the time of value of time \n");
  scanf ( " %d :%d %s", &t[i].tim.val, &t[i].min . &t[i].tim.pre);
 }
}
void timePrint (struct train t[5])
 {
  for (int i = 0; i <5 , i++)
  {
   if(t[i].tim.val == 12 && strcmpi(t[i].tim.pre, "pm") == 0)
   {
    t[i].tim.val = t[i].tim.val;
   }
   else if (strcmp (t[i].tim.pre, "pm") == 0)
      t[i].tim.val = t[i].tim.val + 12;
   }
 }
```

```
void putData (struct train t[5])
{
    for (int i=0; i<5; i++)
    {
     printf("The name of the station = %s \n", t[i].name);
     printf("The time of the train is = %d:%d Hrs \n", t[i].tim.val,
                                                t[i].tim.min, t[i].tim.pre);
    }
}

int main()
{
 struct train t[5];
 getData (t);
 timePrint (t);
 putData (t);
 return 0;
}
```

OUTPUT

Enter the name of the train 0 = Duronto
Enter the time of value of time :
10 20am

Enter the name of the train 1 = Rajdhani
Enter the time of value of time :
12 25pm

Enter the name of the train 2 = Puri-Intercity
Enter the time of value of time :
5 05 pm

Enter the name of train 3 = Howrah Express
Enter the time of value of time
6 pm

Enter the name of train 4 = Jalrayu
Enter the time of value of time :
6 30 pm

The name of the train = Duronto
The time of the train is = 10:20 Hrs

The name of the train = Rajdhani
The time of the train is = 12:25 Hrs

The name of the train = Puri-Intercity
The time of the train is = 17:05 Hrs

The name of the train = Howrah Express
The time of the train is = 18:00 Hrs

The name of the train = Jalrayu.
The time of the train is = 18:30 H

3. WAP to sort the record of the students. The structure declaration is given as below:

```
struct student {
 char name [20];
 int roll no;
 int marks [5]
 int total;
 char grade;
};
```

Sol^n:

```
# include <stdio.h>
# include <string.h>
# define n 2

struct student {
char name [20];
int roll no;
int marks [5];
int total;
char grade;
};
int main()
{
 struct student s[n], t;
 int i, j, k;
 int a[n];
 printf("Enter the students Detail :- \n");
 for (i=0; i<n; i++)
 {
  printf(" %d) Student name : ", i+1);
  scanf("%s", s[i]. name);
  printf("Roll No. : ");
 scanf("%d", &s[i]. roll no);
  for (j=0; j<5; j++)
```

```
{
  printf("Marks in Subject %d : ", j+1);
  scanf("%d", &s[i].marks[j]);
}
s[i].total=(s[i].marks[0]+s[i].marks[1]+s[i].marks[2]+s[i].marks[3]+
                                                    s[i].marks[4]);

printf(" Total Marks : %d\n", s[i].total);
k= s[i].total /5;
if(k>=90)
s[i].grade ='O';
else if(k>=80 && k<90)
s[i].grade = 'E';
else if(k>=70 && k<80)
s[i].grade = 'A';
else if(k>=60 && k<70)
s[i].grade = 'B';
else if(k>=50 && k<60)
s[i].grade = 'C';
else if(k>=40 && k<50)
s[i].grade = 'D';
else
s[i].grade = 'F';
printf("Grade : %c\n", s[i].grade);
}
for(i=0; i<n; i++)
{
  for(j=0; j<n-1; j++)
  { if (s[j].total<s[j+1].total)
```

```
        {
          t = s[j];
          s[j] = s[j+1];
          s[j+1] = t;
        }
    }
}.
printf("\n\n Sorting on the basis of total marks obtained :- \n");
printf("\n\n Roll No. \t Name \t Total Marks \t Grade \n");
for (i=0; i<n; i++)
{
printf("%d \t \t %s \t %d \t \t %c \n", s[i].rollno, s[i].name, s[i].total,
                                        s[i].grade);
}

}
```

OUTPUT

Enter the students Detail :-

1) Student Name : Devansh

Roll No : 10

Marks in Subject 1 : 89

Marks in Subject 2 : 89

Marks in Subject 3 : 89

Marks in Subject 4 : 90

Marks in Subject 5 : 90

Total Marks : 447

Grade : E

2) Student Name : Suresh
Roll No : 29
Marks in Subject 1: 90
Marks in Subject 2: 90
Marks in Subject 3: 99
Marks in Subject 4: 90
Marks in Subject 5 : 90
Total Marks : 459
Grade : O

3) Student Name : Akash
Roll No : 42
Marks in Subject 1 : 99
Marks in Subject 2 : 99
Marks in Subject 3 : 90
Marks in Subject 4 : 99
Marks in Subject 5 : 99
Total Marks : 486
Grade : O

OUTPUT

Sorting on the basis of total marks obtained :-

Roll No.	Name	Total Marks	Grade
42	Akash	486	O
29	Suresh	459	O
10	Devansh	447	E

Q5. WAP to multiply two matrices of order $m \times n$ and $n \times k$.

Solⁿ:

```
#include <stdio.h>
void getMatrixElements (int matrix [][10], int row, int column)
{
 int i, j;
 printf("\n Enter elements : \n");
 for (i=0; i < row; ++i)
 {
  for (j=0; j< column; ++j)
  {
  printf("Enter a %d%d: ", i+1, j+1);
  scanf("%d", &matrix [i][j];
  }
 }
}
void multiplyMatrices (int first [][10], int second [][10], int result [][10],
                                                int r1, int c1, int r2, int c2)
{
 int i, j, k;
 for (i=0; i<r1; ++i)
 {
  for (j=0; j<c2; ++j)
  {
    result [i][j] = 0;
  }
 }
 for (i=0; i<r1; ++i)
 {for (j=0; j<c2; ++j)
  {for (k=0; k<c1; ++k)
```

```
                result[i][j] += first[i][k] * second[k][j];
            }
        }
    }
}
void display(int result[][10], int row, int column)
{
    int i, j;
    printf("\n Output Matrix :\n");
    for (i=0; i<row; ++i)
    {
        for (j=0; j<column; ++j)
        {
            printf("%d", result[i][j]);
            if (j == column-1)
                printf("\n");
        }
    }
}
int main()
{
    int first[10][10], second[10][10], result[10][10], r1, c1, r2, c2;
    printf("Enter rows and columns for first matrix : ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for second matrix : ");
    scanf("%d %d", &r2, &c2);
```

```
while (c1 != r2)
{
  printf("Error! Enter rows and columns again. \n");
  printf("Enter rows and colums for the first matrix : ");
  scanf("%d %d", &r1, &c1);
  printf("Enter rows and column for the second matrix : ");
  scanf("%d %d", &r2, &c2);
}

getMatrixElements (first, r1, c1);
getMatrixElements (second, r2, c2);
multiplyMatrix (first, second, result, r1, c1, r2, c2);
display (result, r1, c2);
return 0;
}
```

OUTPUT

```
Enter rows and column for the first matrix :
Enter rows and column for the second matrix :

Enter elements :
Enter a11 : 1
Enter a12 : 2
Enter a21 : 3
Enter a22 : 4

Enter elements :
Enter a11 : 5
Enter a12 : 6
Enter a13 : 7
Enter a21 : 8
Enter a22 : 9
Enter a23 : 2

Output Matrix :
21  24  9
47  54  25.
```

6. WAP to print the magic matrix.

Solⁿ

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

#define S 20
int main()
{
int order, i, j, k, p, q, mid, matrix [S][S];
 printf("Enter order of magic square (order must be odd): ");
 scanf("%d", &order);
 if (order % 2 == 0)
{
 printf("order must be odd.\n");
 printf(" Exiting ... ");
 exit(0);
}

 mid = order/2;
 for (i=0; i< order; i++)
 {
 for (j=0; j< order; j++)
  {  matrix [i][j]=0;
  }
 }
 k= mid;
 j = order- 1;
 for (i=1; i<= order * order; i++)
 {   matrix [j][k] = i;
     p= j++;
     q = k--;
```

```
    if (j > order - 1)
    {j = 0;
    }
  if (k < 0)
   {
     k = order - 1;
   }
  if (matrix[j][k] != 0)
   {
     k = q;
     j = p-1;
     }
}
printf("\n Obtained Magic Square is : \n");
for(j = 0; j < order; j++)
 {
   for (k = 0; k < order; k++)
   {printf("%4d \t", matrix[j][k]);
   }
   printf("\n");
 }
return 0;
}
```

OUTPUT

```
Enter order of magic square (order must be odd): 3

Obtained Magic Square is :
 2    9    4
 7    5    3
 6    1    8
```

4. Given the following structures:

```
struct date {
int day;
int month;
int year;
};
struct employee{
char name[20];
struct date dob;
struct date doj;
int salary;
}
```

WAP to sort the record

(i) By name
(ii) By date of birth
(iii) By date of joining
(iv) By salary.

```
# include <stdio.h>
# include <stdlib.h>
# include <string.h>
struct date {
 int day;
 int month;
 int year;
};
struct employee{
char name[20];
struct date dob;
struct date doj;
 int salary;
};
```

```
int main().
{
  int i, j, n;
  struct employee e[20], t, k;
printf(" Enter the no. of employee : ");
scanf(" %d", &n);
printf("\n Enter employee details :- \n");
for (i=0; i<n; i++)
{
  printf("\n %d] Name : ", i+1);
  scanf("%s", e[i].name);
  printf(" Date of Birth : ");
scanf("%d %d %d", &e[i].dob.day, &e[i].dob.month, &e[i].dob.
                                                  year);
  printf(" Date of Joining : ");
scanf(" %d %d %d", &e[i].doj.day, &e[i].doj.month, &e[i].doj.year);
printf(" Salary : ");
scanf(" %d", &e[i].salary);
}
for (i=0; i<n; i++)
{
  for (j=0; j<n-1-i; j++)
  {
    if (strcmp(e[j].name, e[j+1].name)>0)
    {
      k=e[j];
      e[j]=e[j+1];
      e[j+1]=k
    }
} }
}
```

```
// display the sorted array
printf("\n Arranged alphabetically : \n");
printf("S.N \t NAME \t DATE OF BIRTH \t JOINING DATE \t SALARY \n");
for(i=0; i<n; i++)
{
printf("%d) \t %s \t %d || %d || %d \t %d || %d || %d \t %d \n", i+1,
        e[i].name, e[i].dob.day, e[i].dob.month, e[i].dob.year,
  e[i].doj.day, e[i].doj.month, e[i].doj.year, e[i].salary);
}
for (i=0; i<n; i++)
{
  for(j=0; j<n-1; j++)
  {
    if (e[j].dob.year < e[j+1].dob.year)
    {
      k= e[j];
      e[j]= e[j+1]
      e[j+1]=k;
    }
    else if(e[j].dob.year == e[j+1].dob.year)
    {
      if(e[j].dob.month < e[j+1].dob.month)
      {
        k= e[j];
        e[j]= e[j+1];
        e[j+1]= k;
      }
```

```
else if (e[j].dob.day < e[j+1].dob.day)
{
        k= e[j]
        e[j]= e[j+1];
        e[j+1]=k;
    }
  }
 }
}

printf("\n Arranged on the basis of Date of Birth: \n");
printf("S.N \t NAME \t DATE OF BIRTH \t JOINING DATE \t SALARY \n");
for (i=0; i<n, i++)
{
printf("%d) \t %s \t %d \ %d \ %d \ %d \ %d \t %d \n", i+1.name, e[i].dob.day,
  e[i].dob.month, e[i].dob.year, e[i].doj.day, e[i].doj.month,
  e[i].doj.year, e[i].salary);
}
for (i=0; i<n; i++)
{
 for (j=0; j<n-1; j++)
 {
  if (e[j].doj.year < e[j+1].doj.year)
  {
     k= e[j]
     e[j]= e[j+1];
     e[j+1]= k;
  }
```

```
    else if (e[j].doj.year == e[j+1].doj.year)
{
  if (e[j].doj.month < e[j+1].doj.month)
  {
     k = e[j]
    e[j] = e[j+1];
    e[j+1] = k;
  }
  else if (e[j].doj.day < e[j+1].doj.day).
  {
     k = e[j];
     e[j] = e[j+1];
    e[j+1] = k;
  }
    }
  }
}
printf("%d)\t %s \t %d\\ %d\\ %d\ t %d\\ %d\\ %d\\ %d\\ %d \t %d \n",
 i+1, e[i].name, e[i].dob.day, e[i].dob.month, e[i].dob.year, e[i].
doj.day, e[i].doj.month, e[i].doj.year, e[i].salary);
}
for (i=0; i<n; i++)
 {
  for (j=0; j<n-1, j++)
  { if (e[j].salary < e[j+1].salary)
    {
```

```
            k= e[j];
            e[j]= e[j+1];
            e[j+1] = k;
        }
    }
}
printf("\n Arranged on the basis of Salary: \n");
printf(" S.N \t NAME \t DATE OF BIRTH \t JOINING DATE \t SALARY \n");
for(i=0; i<n; i++)
{
printf(" %d) \t %s \t %d \ %d \t %d \ %d \t %d \n", i+1, e[i].name,
    e[i].dob.day, e[i].dob.month, e[i].dob.year, e[i].doj.day,
    e[i].doj.month, e[i].doj.year, e[i].salary);
}

}
```

OUTPUT

Enter the no. of employees:

Enter employee detail :-

1) Name: Rajan

Date of Birth: 29
12
78

Date of Joining: 07
07
99

Salary: 7600.

2) Name : Dogeshwar
Date of Birth : 29
09
70
Date of Joining : 19
04
90
Salary : 8900

3) Name : Raju
Date of Birth : 21
11
91
Date of Joining : 25
05
15
Salary : 69000

Arranged alphabetically :

S.N	NAME	DATE OF BIRTH	JOINING DATE	SALRY.
1)	Dogeshwar	29/9/70	19/04/90	8900
2)	Raju	31/11/91.	25/05/15	69000.
3)	Sajan.	29/12/78	7\7\99	7600

Arrang, on the basis of Date of Birth :

S.N	NAME	DATE OF BIRTH	JOINING DATE	SALARY
1)	Raju	31\11\91	25/05/15	69000
2)	Sajan	29\12\78	7\7\99	7600
3)	Dogeshwar	29\9\70	19\04\90	8900

Arrange on the basis of Joining Date :

S.N	NAME	DATE OF BIRTH	JOINING DATE	SALARY
1)	Sajan	29\12\78	7\7\99	7600
2)	Dogeshwar	29\9\70	19\04\90	8900.
3)	Raju	31\11\91	25\05\15	69000

Arranged on the basis of salary:

S.N	NAME	DATE OF BIRTH	JOINING DATE	SALARY
1)	Raju	81\11\91.	25\05\15	69000
2)	Dogeshwor	29\9\70.	19\4\90	8900
3)	Bajan.	29\12\78	7\7\99	7600.