**Name – Akshat Jaiswal**

**Roll No. – 21052646**

**Section – CSE 37**

**DSA LAB 2**

1. **WAP to find the largest number and count the occurrence of the largest number in a dynamic array of n integers using a single loop.**

**Input:**

#include<stdio.h>

#include <stdlib.h>

*int* main()

{

*int* n;

    printf("Enter no of elements: ");

    scanf("%d", &n);

*int* \*ptr;

    ptr = (*int*\*)malloc(n \* sizeof(*int*));

    for(*int* i = 0; i < n; i++)

    {

        printf("Enter element %d : ", (i+1));

        scanf("%d", (ptr+i));

    }

*int* max = \*ptr, count = 0;

    for (*int* i = 0; i < n; i++)

    {

       if(\*(ptr+i) > max)

       {

           max = \*(ptr+i);

           count = 0;

       }

       if(\*(ptr+i) == max)

       {

        count++;

       }

    }

    free(ptr);

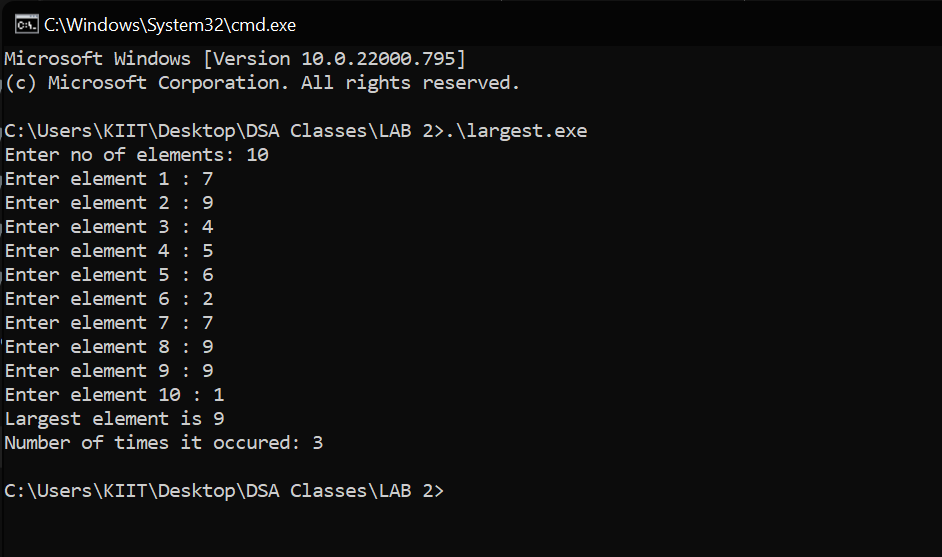
    printf("Largest element is %d \n", max);

    printf("Number of times it occured: %d \n", count);

    return 0;

}

**Output:**

****

1. **Given a dynamic array, WAP to print the next greater element (NGE) for every element. The next greater element for an element x is the first greater element on the right side of x in array. Elements for which no greater element exist, consider next greater element as -1. E.g. For the input array [2, 5, 3, 9, 7], the next greater elements for each elements are as follows.**

**Input:**

#include<stdio.h>

#include <stdlib.h>

*int* main()

{

*int* n,nge=-1;

    printf("Enter no of elements: ");

    scanf("%d", &n);

*int* arr[n];

*int* \*ptr;

    ptr = (*int*\*)malloc(n \* sizeof(*int*));

     if(ptr==NULL)

    {

        printf("\nMemory not available!");

        exit(1);

    }

    for(*int* i = 0; i < n; i++)

    {

        printf("Enter element %d : ", (i+1));

        scanf("%d", &arr[i]);

    }

    printf("Element\tNGE\n");

    for(*int* i=0;i<n;i++)

    {

        nge=-1;

        for(*int* j=i+1;j<n;j++)

        {

            if(arr[j]>arr[i])

            {

                nge=arr[j];

                break;

            }

        }

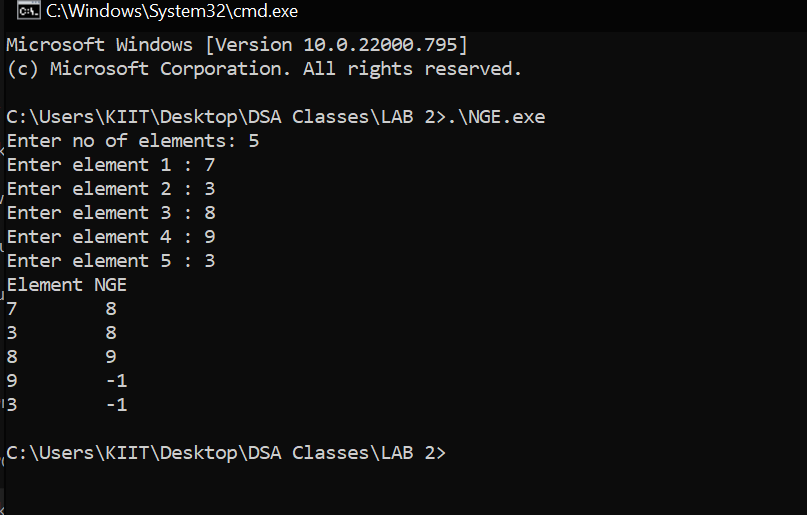
        printf("%d \t %d\n",arr[i],nge);

    }

    return 0;

}

**Output:**

****

1. **WAP to store n student’s information (i.e. student’s roll no, name, gender, marks etc) of an educational institute and display all the data, using array of structure.**

**Input:**

#include <stdio.h>

#include <stdlib.h>

*struct* student

{

*char* name[100];

*char* gender[10];

*int* eng\_marks, math\_marks, phy\_marks, chem\_marks, comp\_marks;

*int* roll;

};

*int* main(*void*)

{

*int* n, i;

*float* total[100];

  printf("Enter number of students: ");

  scanf("%d", &n);

*struct* student stu[n];

  for (i = 0; i < n; i++)

  {

    printf("\nEnter the student %d name : ", i + 1);

    scanf("%s", &stu[i].name);

    fflush(stdin);

    printf("\nEnter the student %d gender : ", i + 1);

    scanf("%s", &stu[i].gender);

    fflush(stdin);

    printf("\nEnter the student %d roll no. : ", i + 1);

    scanf("%d", &stu[i].roll);

    fflush(stdin);

    printf("\nEnter the student %d marks in 5 subjects: ", i + 1);

    printf("\nEnglish: ");

    scanf("%d", &stu[i].eng\_marks);

    printf("\nMaths: ");

    scanf("%d", &stu[i].math\_marks);

    printf("\nPhysics: ");

    scanf("%d", &stu[i].phy\_marks);

    printf("\nChemistry: ");

    scanf("%d", &stu[i].chem\_marks);

    printf("\nComputer: ");

    scanf("%d", &stu[i].comp\_marks);

    total[i] += stu[i].eng\_marks + stu[i].math\_marks + stu[i].phy\_marks + stu[i].chem\_marks

    + stu[i].comp\_marks;

  }

  printf("\n");

  for (i = 0; i < n; i++){

    printf("\n\nStudent %d details: \n", i + 1);

    printf("\nName: %s\n", stu[i].name);

    printf("\nGender: %s\n", stu[i].gender);

    printf("\nRoll NO.: %d\n", stu[i].roll);

    printf("\n\_\_\_\_\_\_\_\_Marks\_\_\_\_\_\_\_\_\n");

    printf("English\tMaths\tPhysics\tChemistry\tComputer\n");

    printf("%d\t%d\t %d\t  %d\t\t%d\n", stu[i].eng\_marks, stu[i].math\_marks, stu[i].phy\_marks,

     stu[i].chem\_marks, stu[i].comp\_marks);

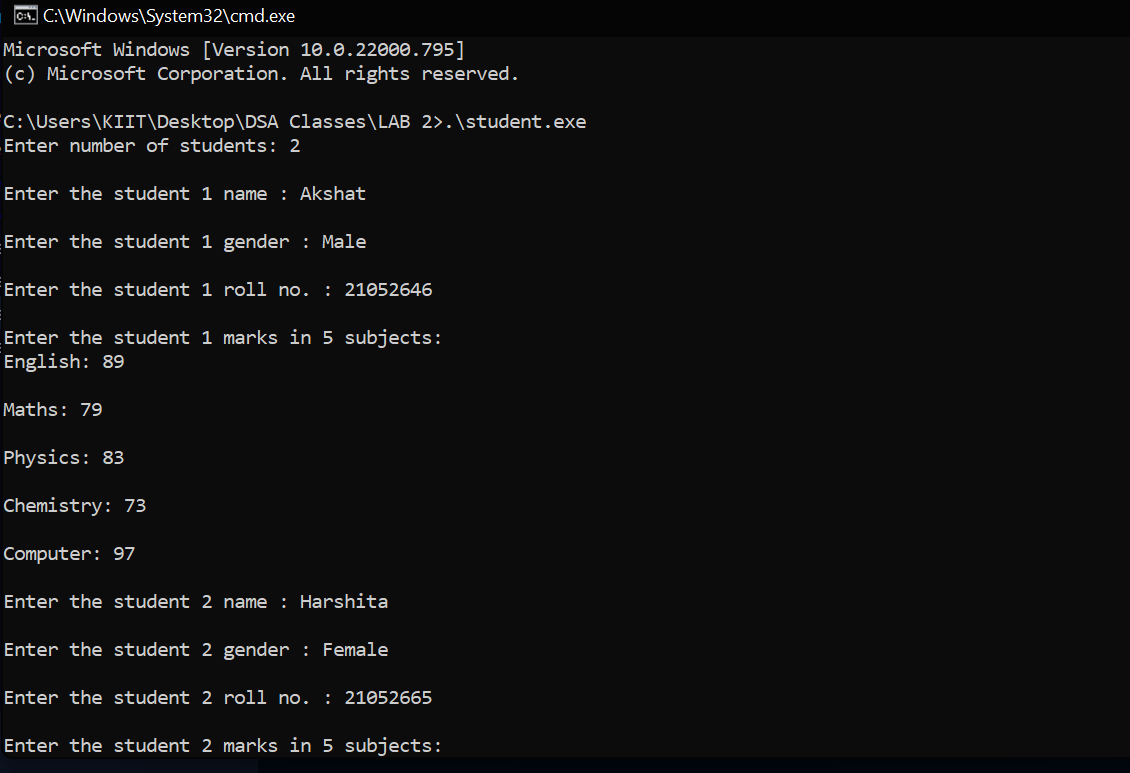
    printf("\nTotal Marks: %.2f\n", total[i]);

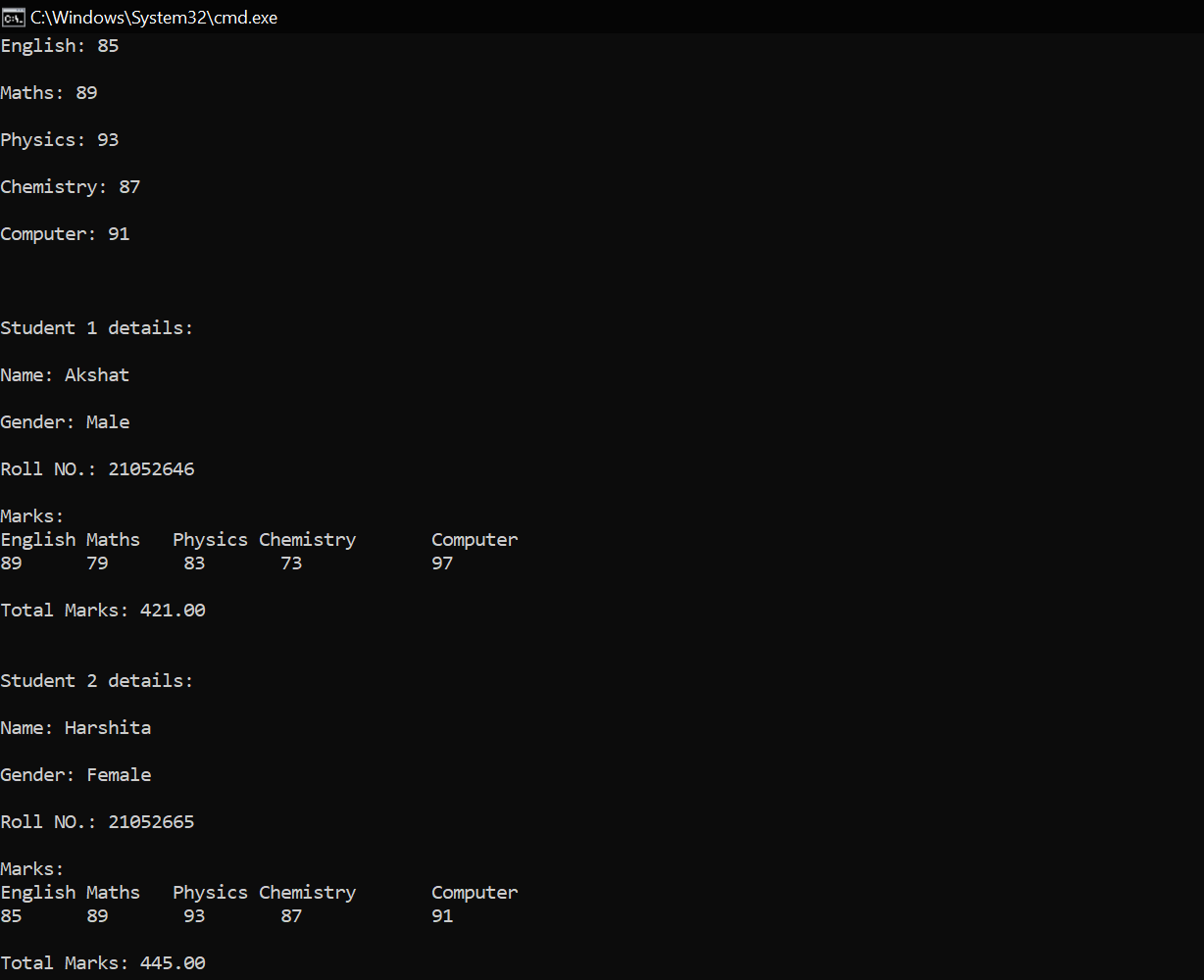
  }

  return 0;

}

**Output:**

****

****

1. **WAP to store n employee’s data such as employee name, gender, designation, department, basic pay. Calculate the gross pay of each employees as follows:**

**Gross pay = basic pay + HR + DA**

**HR=25% of basic and DA=75% of basic**

**Input:**

#include <stdio.h>

#include <stdlib.h>

*struct* employee{

*char* name[100];

*char* gender[100];

*char* designation[100];

*char* department[100];

*int* basic\_pay;

};

*int* main (*void*)

{

*int* n,i;

*float* hr[100],da[100];

  printf("Enter number of employees: ");

  scanf("%d",&n);

*int* \*ptr=(*int*\*) calloc(n,sizeof(*int*));

    if(ptr==NULL)

    {

        printf("\nMemory not available!");

        exit(1);

    }

*struct* employee emp[n];

  for ( i = 0; i < n; i++)

  {

    printf("Enter the employee %d name : ",i+1);

    scanf("%s",&emp[i].name);

    fflush(stdin);

    printf("Enter the employee %d gender : ",i+1);

    scanf("%s",&emp[i].gender);

    fflush(stdin);

    printf("Enter the employee %d designation : ",i+1);

    scanf("%s",&emp[i].designation);

    fflush(stdin);

    printf("Enter the employee %d department : ",i+1);

    scanf("%s",&emp[i].department);

    fflush(stdin);

    printf("Enter the employee %d salary : ",i+1);

    scanf("%d",&emp[i].basic\_pay);

    fflush(stdin);

    hr[i]=(emp[i].basic\_pay)\*0.25;

    da[i]=emp[i].basic\_pay\*0.75;

  }

  printf("\n");

for ( i = 0; i < n; i++)

{

    printf("\n\nEmployee %d details: \n",i+1);

    printf("\nName: %s\n",emp[i].name);

    printf("\nGender: %s\n",emp[i].gender);

    printf("\nDesignation: %s\n",emp[i].designation);

    printf("\nDepartment: %s\n",emp[i].department);

    printf("\nSalary: %d\n",emp[i].basic\_pay);

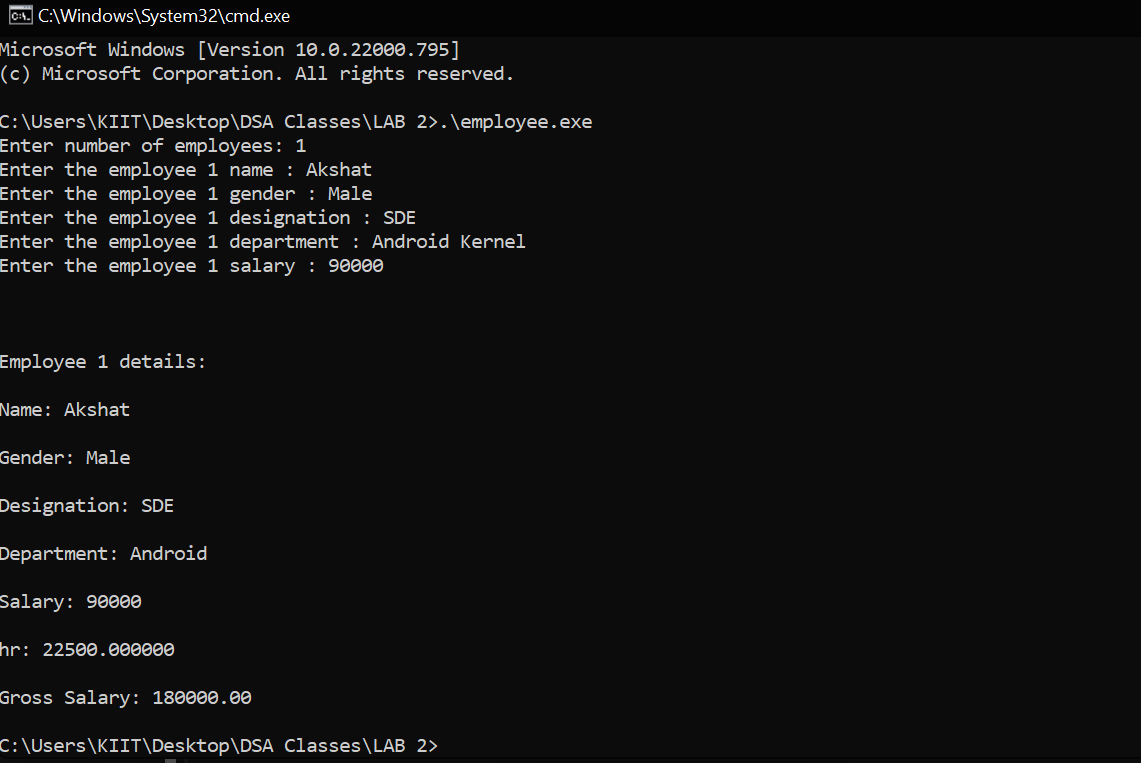
    printf("\nhr: %f\n",hr[i]);

    printf("\nGross Salary: %.2f\n",(emp[i].basic\_pay+hr[i]+da[i]));

}

return 0;

}

**Output:** ****

1. **WAP to declare one distance structure (with members kilometer and meter) and create the variables for addition of two distances using Pointers to structure.**

**Input:**

#include <stdio.h>

*struct* Distance {

*float* kilometer;

*float* meter;

} d1, d2, result;

*int* main() {

   printf("Enter 1st distance\n");

   printf("Kilometer: ");

   scanf("%f", &d1.kilometer);

   fflush(stdin);

   printf("Meter: ");

   scanf("%f", &d1.meter);

   printf("\nEnter 2nd distance\n");

   printf("Kilometer: ");

   scanf("%f", &d2.kilometer);

   fflush(stdin);

   printf("Meter: ");

   scanf("%f", &d2.meter);

   result.kilometer = d1.kilometer + d2.kilometer;

   result.meter = d1.meter + d2.meter;

   printf("\nSum of kilometers = %.2f km\n", result.kilometer);

   printf("\nSum of meters = %.2f m\n", result.meter);

   result.meter=result.meter/1000;

   if (result.meter >= 1000)

   {

      result.meter = result.meter/1000;

      result.kilometer++;

   }

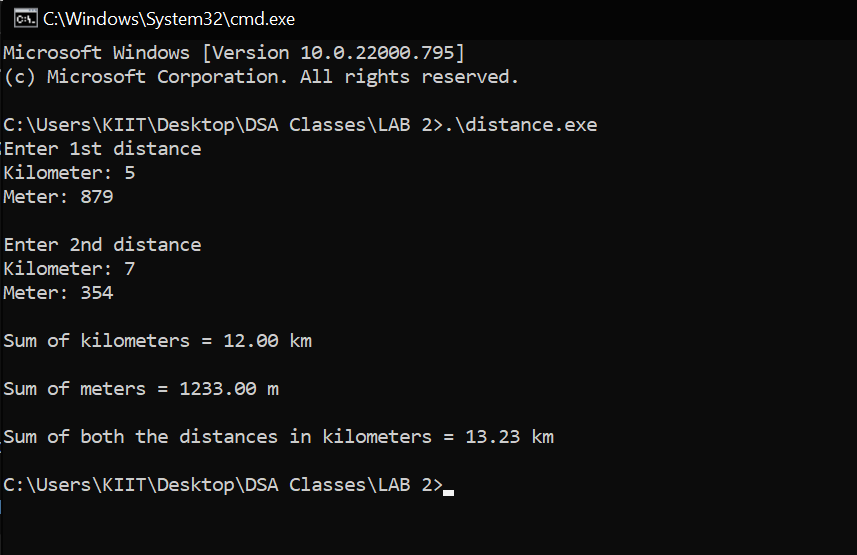
*float* sum=result.kilometer + result.meter;

   printf("\nSum of both the distances in kilometers = %.2f km\n", sum);

   return 0;

}

**Output:**

****