

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

[Deemed to be University](https://en.wikipedia.org/wiki/Kalinga_Institute_of_Industrial_Technology)

**School of Computer Science and Engineering**

Assignment-II

**Submitted By:-**

**Name:** Abhishikt Mohanty **Roll No:** 21052991

**Section:** CSE-40 **Branch:** Computer Science and Engineering **Subject:** DSA LAB

**Assignment 2 | 02/08/2022 | DSA Lab**

Q1. WAP to store n employees 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, DA=75% of basic.

Code :-

#include <stdio.h>

#include <stdlib.h>

struct employee

{

    char name[100];

    char gender[20];

    char designation[100];

    char department[100];

    int basic\_pay;

    int gross\_pay;

};

void gross(struct employee emp[], int n)

{

    int i;

    for (i = 0; i < n; i++)

    {

        emp[i].gross\_pay = 0;

        int hr = (25 \* emp[i].basic\_pay) / 100;

        int da = (75 \* emp[i].basic\_pay) / 100;

        emp[i].gross\_pay = emp[i].basic\_pay + hr + da;

    }

}

int main()

{

    int n;

    printf("Enter the number of employess : ");

    scanf("%d", &n);

    struct employee emp[n];

    int i;

    for (i = 0; i < n; i++)

    {

        printf("Enter the details of employee-%d:\n", i+1);

        printf("Enter the Name : ");

        scanf("%s", &emp[i].name);

        printf("Enter the Gender : ");

        scanf("%s", &emp[i].gender);

        printf("Enter the Designation : ");

        scanf("%s", &emp[i].designation);

        printf("Enter the Department : ");

        scanf("%s", &emp[i].department);

        printf("Enter the Basic Salary : ");

        scanf("%d", &emp[i].basic\_pay);

        printf("\n");

    }

    gross(emp, n);

    for (i = 0; i < n; i++)

    {

        printf("\nDetails of employee-%d\n", i+1);

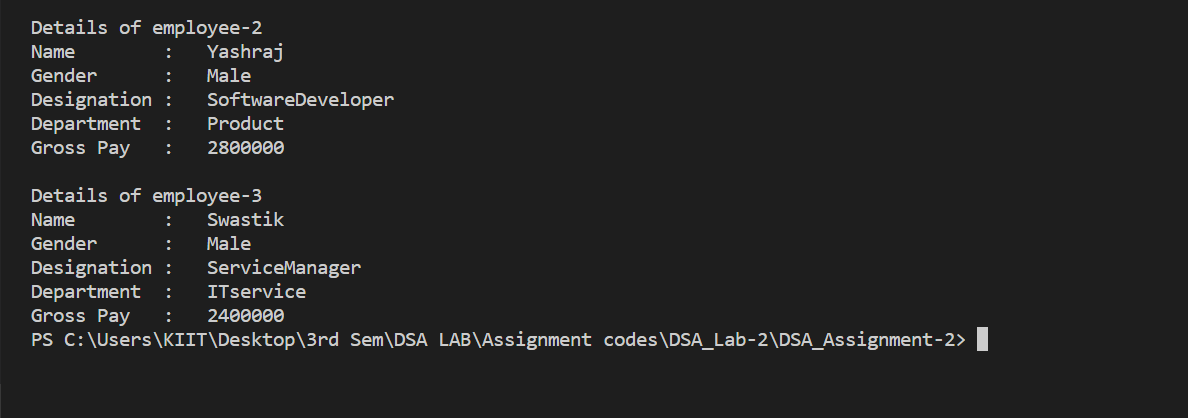
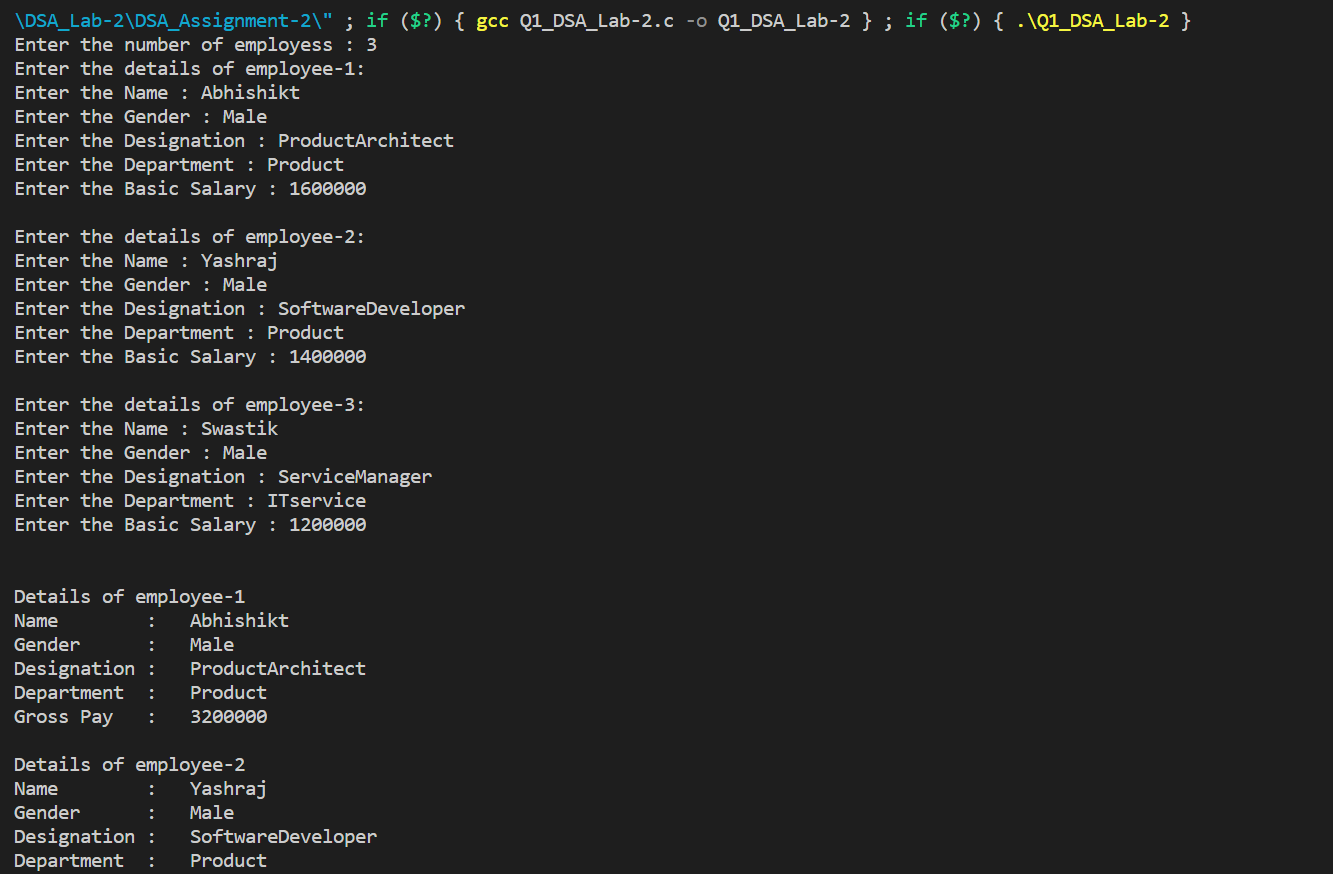
        printf("Name        :\t%s\nGender      :\t%s\nDesignation :\t%s\nDepartment  :\t%s\nGross Pay   :\t%d\n", emp[i].name, emp[i].gender, emp[i].designation, emp[i].department, emp[i].gross\_pay);

    }

    return 0;

}

Output:-



Q2. WAP to add two distances (in km-meter) by passing structure to a function.

Code :-

#include <stdio.h>

struct Distance

{

  int km;

  float mtr;

} firstDistance, secondDistance, sum;

int main()

{

  printf("Enter kilo-meter and meter(up to 2 digit places) for the first distance with a space : ");

  scanf("%d %f", &firstDistance.km, &firstDistance.mtr);

  printf("Enter kilo-meter and meter(up to 2 digit places) for the second distance with a space : ");

  scanf("%d %f", &secondDistance.km, &secondDistance.mtr);

  sum.km = firstDistance.km + secondDistance.km;

  sum.mtr = firstDistance.mtr + secondDistance.mtr;

  while (sum.mtr >= 100)

  {

    sum.mtr = sum.mtr - 100;

    sum.km++;

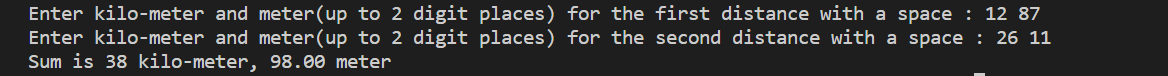
  }

  printf("Sum is %d kilo-meter, %.2f meter\n", sum.km, sum.mtr);

  return 0;

}

Output:-



Q3. Add two complex numbers by passing structures to a function

Code:-

#include <stdio.h>

typedef struct complex

{

    float r;

    float i;

} complex;

complex addition(complex num1, complex num2);

int main()

{

    complex num1, num2, value;

    printf("Enter real and imaginary parts of first complex number: ");

    scanf("%f %f", &num1.r, &num1.i);

    printf("Enter real and imaginary parts of second complex number: ");

    scanf("%f %f", &num2.r, &num2.i);

    value = addition(num1, num2);

    printf("Addition of given complex number is %.1f + %.1fi", value.r, value.i);

    return 0;

}

complex addition(complex num1, complex num2)

{

    complex temp;

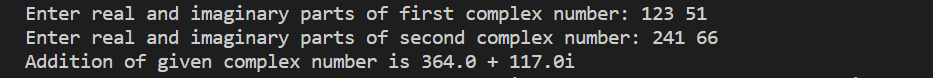
    temp.r = num1.r + num2.r;

    temp.i = num1.i + num2.i;

    return (temp);

}

Output:-



Q4. Calculate the difference between two time periods using structure

Code:-

#include<stdio.h>

struct time

{

    int hours;

    int minutes;

    int seconds;

};

int main()

{

    struct time first, second, diff;

    printf("Enter the first time in 24 hour format:\n");

    printf("Enter hours, minutes and seconds respectively(hh mm ss): ");

    scanf("%d %d %d", &first.hours,&first.minutes, &first.seconds);

    printf("Enter the second time in 24 hour format:\n");

    printf("Enter hours, minutes and seconds respectively(hh mm ss): ");

    scanf("%d %d %d", &second.hours,&second.minutes, &second.seconds);

    if(first.seconds > second.seconds)

    {

        second.seconds += 60;

        --second.minutes;

    }

    if(first.minutes > second.minutes)

    {

        second.minutes += 60;

        --second.hours;

    }

    diff.seconds = second.seconds - first.seconds;

    diff.minutes = second.minutes - first.minutes;

    diff.hours = second.hours - first.hours;

    printf("\nTIME PERIOD : %d:%d:%d - ", first.hours, first.minutes, first.seconds);

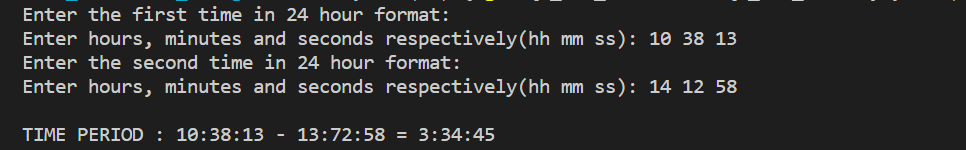
    printf("%d:%d:%d ", second.hours, second.minutes, second.seconds);

    printf("= %d:%d:%d\n", diff.hours, diff.minutes, diff.seconds);

    return 0;

}

Output:-



Q5. Store information of n students using structures and Dynamic Memory Allocation.

Code:-

#include <stdio.h>

#include <stdlib.h>

struct student

{

  char name[100];

  int roll\_number;

  int class;

  char section;

};

int main()

{

  int n;

  printf("Enter the number of students : ");

  scanf("%d", &n);

  struct student \*ptr;

  ptr = (struct student \*)malloc(n \* sizeof(struct student));

  for (int i = 0; i < n; i++)

  {

    printf("\nEnter the deatil of student %d\n", i + 1);

    printf("Enter the name : ");

    scanf("%s", &(ptr + i)->name);

    printf("Enter the Roll number : ");

    scanf("%d", &(ptr + i)->roll\_number);

    printf("Enter the class (in number) : ");

    scanf("%d", &(ptr + i)->class);

    printf("Enter the section : ");

    scanf("%s", &(ptr + i)->section);

  }

  printf("\nDetails of the Students: \n");

  printf("\n");

  for (int i = 0; i < n; i++)

  {

    printf("Student %d", i + 1);

    printf("\nName        : %s", (ptr + i)->name);

    printf("\nRoll Number : %d", (ptr + i)->roll\_number);

    printf("\nClass       : %d", (ptr + i)->class);

    printf("\nSection     : %c", (ptr + i)->section);

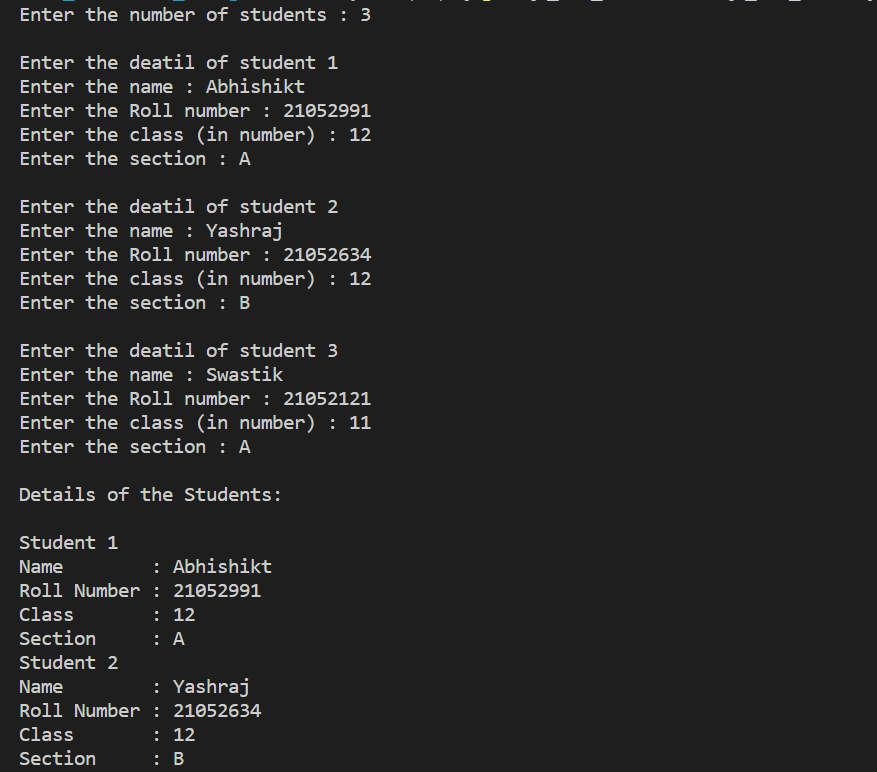
    printf("\n");

  }

  return 0;

}

Output:-



Q6. C program to read a one dimensional array, print sum of all elements along with inputted array elements using Dynamic Memory Allocation.

Code:-

#include <stdio.h>

#include <stdlib.h>

struct array

{

    int a;

    int sum;

};

int main()

{

    int n;

    printf("Enter the length of the array : ");

    scanf("%d", &n);

    struct array \*ptr;

    ptr = (struct array \*)malloc(n \* sizeof(struct array));

    printf("Enter the elements of the array : ");

    for (int i = 0; i < n; i++)

    {

        scanf("%d", &(ptr + i)->a);

    }

    ptr->sum = 0;

    for (int i = 0; i < n; i++)

    {

        ptr->sum += (ptr + i)->a;

    }

    printf("Elements of the array : ");

    for (int i = 0; i < n; i++)

    {

        printf("%d  ", (ptr + i)->a);

    }

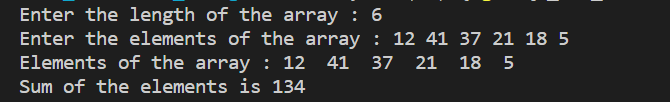
    printf("\n");

    printf("Sum of the elements is %d", ptr->sum);

    return 0;

}

Output:-



Q7. WAP using C to Evaluate the Given Polynomial Equation f(x). Note: Order of polynomial, co-efficient and value of x will be user input.

Code:-

#include <stdio.h>

#include <stdlib.h>

#define MAXSIZE 10

void main()

{

  int a[MAXSIZE];

  int i, N, power;

  float x, polySum;

  printf("Enter the order of the polynomial\n");

  scanf("%d", & N);

  printf("Enter the value of x\n");

  scanf("%f", & x);

  printf("Enter %d coefficients\n", N + 1);

  for (i = 0; i <= N; i++) {

    scanf("%d", & a[i]);

  }

  polySum = a[0];

  for (i = 1; i <= N; i++) {

    polySum = polySum \* x + a[i];

  }

power = N;

    printf("Given polynomial is: \n");

    for (i = 0; i <= N; i++)

    {

        if (power < 0)

        {

            break;

        }

        if (a[i] > 0 & i!=0)

            printf(" + ");

        else if (a[i] < 0)

            printf(" - ");

        else

            printf(" ");

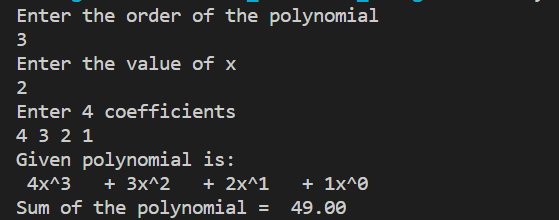
        printf("%dx^%d  ", abs(a[i]), power--);

  }

  printf("\nSum of the polynomial = %6.2f\n", polySum);

}

Output:-



Q8. WAP using function that adds given two polynomials f(x) = h(x) + g(x)

Code:-

Output:-

Q9. WAP to check whether the given matrix is sparse matrix or not.

Code:-

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int row, col, i, j, a[10][10], count = 0;

    printf("Enter number of row: ");

    scanf("%d", &row);

    printf("Enter number of Column: ");

    scanf("%d", &col);

    printf("Enter element of the Matrix: \n");

    for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)

        {

            scanf("%d", &a[i][j]);

        }

    }

    printf("Elements are:\n");

    for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)

        {

            printf("%d\t", a[i][j]);

        }

        printf("\n");

    }

    for (i = 0; i < row; i++)

    {

        for (j = 0; j < col; j++)

        {

            if (a[i][j] == 0)

                count++;

        }

    }

    if (count > ((row \* col) / 2))

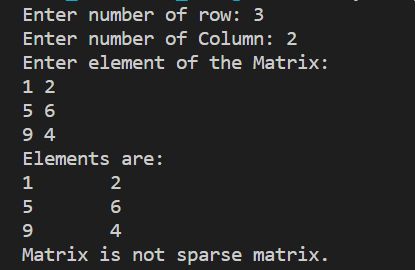
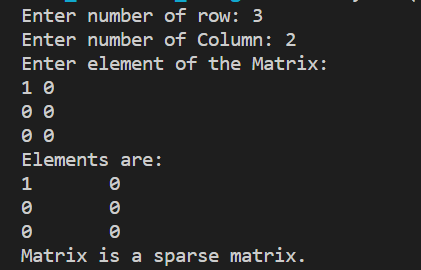
        printf("Matrix is a sparse matrix.\n");

    else

        printf("Matrix is not sparse matrix.\n");

}

Output:- Output:-

**Home-Assignment 2 | 02/08/2022 | DSA Lab**

Q1.

Code:-

Output:-

Q2.

Code:-

Output:-

Q3.

Code:-

Output:-

Q4.

Code:-

Output:-

Q5. WAP to sort rows of a matrix having m rows and n columns in ascending & columns in descending order.

Code:-

Output:-