|  |
| --- |
| **Faculty In charge: JANANI T** |
| **Class No.: VL2021220506935** |

Logo

Description automatically generated

**School of Computer Science and Engineering (SCOPE)**

**Vellore Institute of Technology**

**Vellore.**

|  |
| --- |
|  |
| **BCSE102L - Computer Programming: Python** |
|  |
| **Digital Footprint** |

BCSE101E - Computer Programming: Python

**Table of Contents**

|  |  |  |  |
| --- | --- | --- | --- |
| Ex.No. | Title | Date | Page No. |
| 1 | **M1\_CSQ1** | 21/2/22 | 2 |
| 2 | **M1\_CSQ2** | 21/2/22 | 4 |
| 3 | **M1\_CSQ3** | 21/2/22 | 6 |
| 4 | **M1\_CSQ4** | 21/2/22 | 9 |
| 5 | **M1\_CSQ5** | 21/2/22 | 12 |
| 6 | **M1\_CSQ6** | 25/2/22 | 14 |
| 7 | **M2\_CSQ1** | 28/2/22 | 17 |
| 8 | **M2\_CSQ2** | 28/2/22 | 19 |
| 9 | **M2\_CSQ3** | 28/2/22 | 22 |
| 10 | **M2\_CSQ4** | 3/3/22 | 25 |
| 11 | **M2\_CSQ5** | 3/3/22 | 27 |
| 12 | **M2\_CSQ6** | 3/3/22 | 30 |
| 13 | **M3\_CSQ1** | 14/3/22 | 36 |
| 14 | **M3\_CSQ2** | 14/3/22 | 38 |
| 15 | **M3\_CSQ3** | 14/3/22 | 40 |
| 16 | **M3\_CSQ4** | 18/3/22 | 43 |
| 17 | **M3\_CSQ5** | 19/3/22 | 46 |
| 18 | **M4\_CSQ1** | 24/3/22 | 50 |
| 19 | **M4\_CSQ2** | 24/3/22 | 53 |
| 20 | **M4\_CSQ3** | 25/3/22 | 57 |
| 21 | **M4\_CSQ4** | 7/4/22 | 61 |
| 22 | **M4\_CSQ5** | 7/4/22 | 64 |
| 23 | **M5\_CSQ1** | 21/4/22 | 68 |
| 24 | **M5\_CSQ2** | 22/4/22 | 71 |
| 25 | **M5\_CSQ3** | 22/4/22 | 74 |
| 26 | **M5\_CSQ4** | 22/4/22 | 77 |
| 27 | **M5\_CSQ5** | 27/4/22 | 80 |
| 28 | **M5\_CSQ6** | 27/4/22 | 84 |
| 29 | **M5\_CSQ7** | 27/4/22 | 86 |
| 30 | **M5\_CSQ8** | 27/4/22 | 88 |
| 31 | **M5\_CSQ9** | 27/4/22 | 90 |
| 32 | **M5\_CSQ10** | 5/5/22 | 94 |
| 33 | **M6\_CSQ1** | 5/5/22 | 99 |
| 34 | **M6\_CSQ2** | 5/5/22 | 103 |
| 35 | **M6\_CSQ3** | 5/5/22 | 107 |
| 36 | **M6\_CSQ4** | 6/5/22 | 113 |
| 37 | **M6\_CSQ5** | 6/5/22 | 120 |
| 38 | **M6\_CSQ6** | 19/5/22 | 126 |
| 39 | **M6\_CSQ7** | 19/5/22 | 133 |
| 40 | **M6\_CSQ8** | 19/5/22 | 137 |
| 41 | **M7\_CSQ1** | 23/5/22 | 141 |
| 42 | **M7\_CSQ2** | 23/5/22 | 146 |
| 43 | **M7\_CSQ3** | 26/5/22 | 152 |
| 44 | **M7\_CSQ4** | 26/5/22 | 156 |
| 45 | **M7\_CSQ5** | 1/6/22 | 160 |
| 46 | **M7\_CSQ6** | 1/6/22 | 164 |
| 47 | **M7\_CSQ7** | 1/6/22 | 169 |
| 48 | **M7\_CSQ8** |  | 173 |
| 49 | **M7\_CSQ9** |  |  |
| 50 | **M7\_CSQ10** |  |  |

|  |
| --- |
| **Signature of the student (Digital)** |
| **Abhinav Dinesh Srivatsa** |

## [Ex. No. M1\_CSQ1]

### AIM

Write a c program to read a value and check whether it is "even" or "odd". Only positive value can be accepted, otherwise display "Enter only positive number"

### Algorithm / Pseudocode

Declare integer Num

Read input and store as Num

If Num > 0, then

If Num % 2 = 0, then display ‘even’

Else display ‘odd’

Else display ‘Enter only positive number’

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    int num;

    scanf("%d", &num);

    if (num > 0)

    {

        if (num % 2 == 0)

        {

            printf("even");

        }

        else

        {

            printf("odd");

        }

    }

    else{

        printf("Enter only positive number");

    }

}

**Output**

Graphical user interface, application

Description automatically generated

## [Ex. No. M1\_CSQ2]

### AIM

Given ‘n’ letters find how many isogram words can be formed? A word is said to be isogram if it is formed without repeating a letter. For example, the word ‘isogram’ itself has the property and ‘Apple’ do not have the property as ‘p’ is repeated in the word

### Algorithm / Pseudocode

Declare integer N

Read input and assign to N

Declare integer Fact as 1

Loop from 1 to N as X

Calculate Fact \* X and assign to Fact

Display Fact

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    int n;

    scanf("%d", &n);

    int fact = 1;

    for(int x = 1; x <= n; x++){

        fact \*= x;

    }

    printf("%d", fact);

}

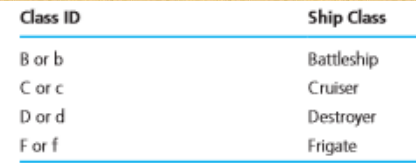
**Output**



## [Ex. No. M1\_CSQ3]

### AIM

Each ship serial number begins with a letter indicating the class of the ship. Write a program that reads a ship’s first character of serial number and displays the class of the ship.



### Algorithm / Pseudocode

Declare character C

Read input and store as C

Declare integer I as C

If I >= 97, then calculate I – 32 and assign to I

Assign C as I

If C = ‘B’, then display ‘Battleship’

If C = ‘C’, then display ‘Cruiser

If C = ‘D’, then display ‘Destroyer’

If C = ‘F’, then display ‘Frigate’

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    char c;

    scanf("%c", &c);

    int i = c;

    if (i >= 97)

    {

        i -= 32;

    }

    c = i;

    if (c == 'B')

    {

        printf("Battleship");

    }

    if (c == 'C')

    {

        printf("Cruiser");

    }

    if (c == 'D')

    {

        printf("Destroyer");

    }

    if (c == 'F')

    {

        printf("Frigate");

    }

}

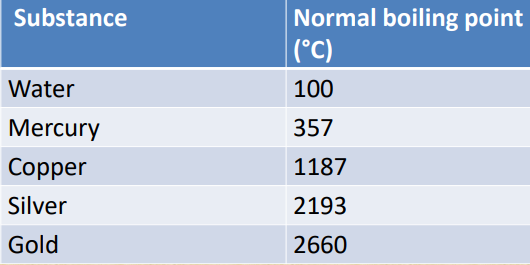
Graphical user interface, application

Description automatically generated**Output**

## [Ex. No. M1\_CSQ4]

### AIM

The table below shows the normal boiling points of several substances. Write a program that prompts the user for the observed boiling point of a substance in °C and identifies the substance if the observed boiling point is within 5% of the expected boiling point. If the data input is more than 5% higher or lower than any of the boiling points in the table, the program should output the message Substance unknown.



### Algorithm / Pseudocode

Declare integer array Temps as array of given temperatures

Declare integer T

Read input and store as T

Declare integer flag as 0

Loop from 0 to length of array Temps as X

Declare integer Lowert as Temps[X] \* 0.95

Declare integer Uppert as Temps[X] \* 1.05

If T > Lowert and T < Uppert, then

If X = 0, then display ‘Water’

If X = 1, then display ‘Mercury

If X = 2, then display ‘Copper

If X = 3, then display ‘Silver’

If X = 4, then display ‘Gold’

Assign Flag as 1

Break the loop

If Flag = 0, then print ‘Substance unknown’

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    int temps[] = {100, 357, 1187, 2193, 2660};

    int t;

    scanf("%d", &t);

    int flag = 0;

    for (int x = 0; x < sizeof(temps) / sizeof(temps[0]); x++)

    {

        int lowert = temps[x] \* 0.95;

        int uppert = temps[x] \* 1.05;

        if (t > lowert && t < uppert)

        {

            switch (x)

            {

            case 0:

                printf("Water");

                break;

            case 1:

                printf("Mercury");

                break;

            case 2:

                printf("Copper");

                break;

            case 3:

                printf("Silver");

                break;

            case 4:

                printf("Gold");

            }

            flag = 1;

            break;

        }

    }

    if (flag == 0)

    {

        printf("Substance unknown");

    }

}

### Output

## [Ex. No. M1\_CSQ5]

### AIM

Check whether a number is prime or not using while statement, otherwise print "Not Prime"

### Algorithm / Pseudocode

Declare integer N

Declare integer flag as 0

Read input and assign to N

Declare integer X as 2

Loop while X < N / 2

If N % X = 0, then display ‘Not Prime’, set Flag as 1 and break loop

Increment X by 1

If Flag is 0, then display ‘Prime’

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    int n;

    int flag = 0;

    scanf("%d", &n);

    int x = 2;

    while (x < n / 2)

    {

        if (n % x == 0)

        {

            printf("Not Prime");

            flag = 1;

            break;

        }

        x++;

    }

    if (flag == 0)

    {

        printf("Prime");

Graphical user interface, application

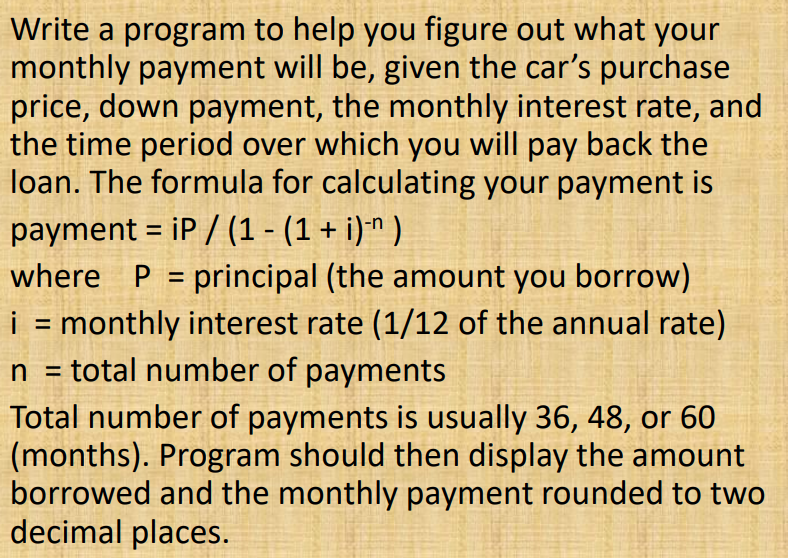
Description automatically generated    }

### Output

## [Ex. No. M1\_CSQ6]

### AIM

Develop a c Program to deal with N loans. Use math.h library for pow(a,b) and printf("%.2f",a) to print a value with two precision.



### Algorithm / Pseudocode

Declare integer N

Read input and assign to N

Declare integers P, D, I, T

Loop for integer X as 0 while less than N

Read 4 inputs and assign them to P, D, I, T respectively

If T = 36, 48 or 60, then

Declare float Loan as P – D

Assign Loan as Loan \* I / 1200

Declare float Den as 1 – (1 + I/1200)-T

Assign Loan as Loan / Den

Display Loan with 2 decimal points

### Program Code

//21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <math.h>

int main()

{

    int n;

    scanf("%d", &n);

    int p, d, i, t;

    for (int x = 0; x < n; x++)

    {

        scanf("%d %d %d %d", &p, &d, &i, &t);

        if (t == 36 || t == 48 || t == 60)

        {

            float loan = p - d;

            loan \*= (float)i / 1200;

            float den = 1 - pow(1 + (float)i/1200, -(float)t);

            loan /= den;

            printf("%.2f\n", floor(loan \* 100) / 100);

        }

    }

}

### Output

Graphical user interface

Description automatically generated with medium confidence

## [Ex. No. M2\_CSQ1]

### AIM

Each year the Department of Traffic Accidents receives accident count reports from several cities and towns across the country. Given details of ‘n’ days, develop an algorithm and write a program to determine the average number of accidents and for each day, print the difference between the number of accidents on that day and average. For example, if the number of accidents is 5 and the values are 10, 12, 15, 13, 5 then average is 11 and the difference of values are 1, 1, 4, 2, 6.

### Algorithm / Pseudocode

Declare integer N

Read input and assign to N

Declare integer array Acc of length N

Declare integer Sum as 0

Loop from 0 to N as X

Read input and assign to Acc[N]

Calculate Sum + Acc[N] and assign to Sum

Declare integer Mean as Sum / N

Loop from 0 to N as X

Calculate absolute value of Acc[N] - Mean and assign to Acc[N]

Display Acc[N]

### Program Code

//21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n;

    scanf("%d", &n);

    int acc[n];

    int sum = 0;

    for (int x = 0; x < n; x++)

    {

        scanf("%d", &acc[x]);

        sum += acc[x];

    }

    int mean = sum / n;

    for (int x = 0; x < n; x++)

    {

        acc[x] = abs(mean - acc[x]);

        printf("%d\n", acc[x]);

    }

}

### Graphical user interface Description automatically generatedOutput

## [Ex. No. M2\_CSQ2]

### AIM

Huffman code is a particular type of optimal prefix code for characters. It is commonly used for lossless data compression. It is a variable-length code derived from frequency of occurrence. Given a string develop an algorithm and write a C program to determine frequency of occurrence of each character in the string.

### Algorithm / Pseudocode

Declare character array as Sen

Read input till new line and assign to Sen

Declare integer array Count

Declare integer X as 0

Loop while X < 26

Assign Count[X] as 0 and increment X

Declare integer Ord

Loop while Sen[X] is not ‘\0’

Calculate Ord as integer casted Sen[X]

If Ord is between ‘A’ and ‘Z’, then make it lower case

If Ord is not space, then increment Count[Ord – ‘a’]

Loop from 0 to 26 as X

If Count[X] is not 0, then display the character spaced with the Count[X]

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    char sen[50];

    scanf("%[^\n]s", sen);

    int count[26];

    int x = 0;

    while (x < 26)

    {

        count[x++] = 0;

    }

    x = 0;

    int ord;

    while (sen[x] != '\0')

    {

        ord = (int)sen[x++];

        if (ord <= 'Z' && ord >= 'A')

        {

            ord += 32;

        }

        if (ord != ' ')

        {

            count[ord - 'a']++;

        }

    }

    for (x = 0; x < 26; x++)

    {

        if (count[x] != 0)

        {

            printf("%c %d\n", x + 'a', count[x]);

        }

    }

}

### Output

Graphical user interface, application

Description automatically generated

## [Ex. No. M2\_CSQ3]

### AIM

Create C program to add two matrices.

### Algorithm / Pseudocode

Declare integers R and C

Read inputs and assign to R and C

Declare integer array Mat1 and Mat2 of dimensions R and C

Loop from 0 to R as X

Loop from 0 to C as Y

Read input and assign to Mat1[X][Y]

Loop from 0 to R as X

Loop from 0 to C as Y

Read input and assign to Mat2[X][Y]

Loop from 0 to R as X

Loop from 0 to C as Y

Display Mat1[X][Y] + Mat2[X][Y]

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

int main()

{

    int r, c;

    scanf("%d %d", &r, &c);

    int mat1[r][c];

    int mat2[r][c];

    for (int x = 0; x < r; x++)

    {

        for (int y = 0; y < c; y++)

        {

            scanf("%d", &mat1[x][y]);

        }

    }

    for (int x = 0; x < r; x++)

    {

        for (int y = 0; y < c; y++)

        {

            scanf("%d", &mat2[x][y]);

        }

    }

    for (int x = 0; x < r; x++)

    {

        for (int y = 0; y < c; y++)

        {

            printf("%d\n", mat1[x][y] + mat2[x][y]);

        }

    }

}

### Output

Chart

Description automatically generated with low confidence

## [Ex. No. M2\_CSQ4]

### AIM

Create C program with a function named swap\_twoNumbers, using call by reference.

### Algorithm / Pseudocode

Void swap\_twoNumbers(int \*V1, int \*V2)

Declare integer Temp as \*V1

Assign \*V1 as \*V2

Assign \*V2 as Temp

Declare integers A, B

Read input and assign to A and B

Call swap\_twoNumbers(A, B)

Display A and B

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

void swap\_twoNumbers(int \*i1, int \*i2)

{

    int temp = \*i2;

    \*i2 = \*i1;

    \*i1 = temp;

}

int main()

{

    int a, b;

    scanf("%d%d", &a, &b);

    swap\_twoNumbers(&a, &b);

    printf("%d\n%d", a, b);

}

### Graphical user interface, application Description automatically generatedOutput

## [Ex. No. M2\_CSQ5]

### AIM

Each year the Department of Traffic Accidents receives accident count reports from several cities and towns across the country. Given details of ‘n’ days, develop an algorithm and write a program to determine the average number of accidents and for each day, print the difference between the number of accidents on that day and average. For example, if the number of accidents is 5 and the values are 10, 12, 15, 13, 5 then average is 11 and the difference of values are 1, 1, 4, 2, 6

Use the following functions signature

void read\_Count(int \*, int);

float find\_Mean(int \*, int);

void print\_Diff(int \*, int, float);

### Algorithm / Pseudocode

Void read\_Count(int \*Array, int N)

Loop from 0 to N as X

Read input and assign to Array[X]

Float find\_Mean(int \*Array, int N)

Declare float Sum as 0

Loop from 0 to N as X

Calculate Sum as Sum + Array[X]

Return Sum / N

Void print\_Diff(int \*Array, int N, float Mean)

Loop from 0 to N as X

Display absolute value of Mean – Array[X]

Declare integer N

Read input and assign to N

Declare integer array Acc with N spaces

Call read\_Count(Acc, N)

Declare float Mean and assign by calling find\_Mean(Acc, N)

Call print\_Diff(Acc, N, Mean)

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

void read\_Count(int \*array, int n)

{

    for (int x = 0; x < n; x++)

    {

        scanf("%d", &array[x]);

    }

}

float find\_Mean(int \*array, int n)

{

    float sum = 0;

    for (int x = 0; x < n; x++)

    {

        sum += array[x];

    }

    return sum / n;

}

void print\_Diff(int \*array, int n, float mean)

{

    for (int x = 0; x < n; x++)

    {

        printf("%d\n", abs((int)mean - array[x]));

    }

}

int main()

{

    int n;

    scanf("%d", &n);

    int acc[n];

    read\_Count(acc, n);

    float mean = find\_Mean(acc, n);

    print\_Diff(acc, n, mean);

Graphical user interface, application

Description automatically generated}

### Output

## [Ex. No. M2\_CSQ6]

### AIM

A picture containing text, antenna

Description automatically generatedTic-tac-toe is a paper-and-pencil game for two players, X and O, who take turns marking the spaces n a 3×3 grid. Player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.

Given the board configuration of the tic tac toe game, determine if the board is in either of the following states: empty, player1 wins, player2 wins, draw or intermediate. The board is said to be in initial state if all the cells contain ‘-1’, player1 uses ‘1’ as his coin and player2 uses ‘2’ as his coin. The game is draw when the board is full and no one has won the game. The game is in intermediate state when no one has won and board is not full

Use the following function signatures

void read\_Board(int ttt[][3]);

int count\_EmptyCell(int ttt[][3]);

int check\_Rowwise(int ttt[][3],int);

int check\_Colwise(int ttt[][3],int);

int check\_Diagonalwise(int ttt[][3],int);

### Algorithm / Pseudocode

Void read\_Board(int TTT[3][3])

Loop from 0 to 3 as X

Loop from 0 to 3 as Y

Read input and assign to TTT[X][Y]

Int count\_EmptyCell(int TTT[3][3])

Declare integer Count as 0

Loop from 0 to 3 as X

Loop from 0 to 3 as Y

If TTT[X][Y] = -1, then increment Count

Return Count

Int check\_Rowwise(int TTT[3][3])

Declare integer Val

Loop from 0 to 3 as X

Assign Val as TTT[X][0]

If TTT[X][1] and TTT[X][2] are equal to Val, then return Val

Return 0

Int check\_Colwise(int TTT[3][3])

Declare integer Val

Loop from 0 to 3 as X

Assign Val as TTT[0][X]

If TTT[1][X] and TTT[2][X] are equal to Val, then return Val

Return 0

Int check\_Diagonalwise(int TTT[3][3])

Declare integer Val

Assign Val as TTT[0][0]

If TTT[1][1] and TTT[2][2] equal Val, then return Val

Assign Val as TTT[0][2]

If TTT[1][1] and TTT[2][0] equal Val, then return Val

Return 0

Declare integer array TTT with 3 rows and 3 columns

Call read\_Board(TTT)

Declare integer Emptycells and assign it by calling count\_EmptyCell(TTT)

If Emptycells = 9, then display ‘Empty’

Else if Emptycells = 0, then

Declare integer Rowwin and assign it by calling check\_Rowwise(TTT)

Declare integer Colwin and assign it by calling check\_Colwise(TTT)

Declare integer Diagwin and assign it by calling check\_Diagonalwise(TTT)

If Rowwin, Colwin and Diagwin are 0, then display ‘Draw’

Else if Rowwin is not 0, then display ‘Player{Rowwin} wins’

Else if Colwin is not 0, then display ‘Player{Colwin} wins’

Else if Diagwin is not 0, then display ‘Player{Diagwin} wins’

Else display ‘Intermediate’

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

void read\_Board(int ttt[3][3])

{

    for (int x = 0; x < 3; x++)

    {

        for (int y = 0; y < 3; y++)

        {

            scanf("%d", &ttt[x][y]);

        }

    }

}

int count\_EmptyCell(int ttt[3][3])

{

    int count = 0;

    for (int x = 0; x < 3; x++)

    {

        for (int y = 0; y < 3; y++)

        {

            if (ttt[x][y] == -1)

            {

                count++;

            }

        }

    }

    return count;

}

int check\_Rowwise(int ttt[3][3])

{

    int val;

    for (int x = 0; x < 3; x++)

    {

        val = ttt[x][0];

        if (ttt[x][1] == val && ttt[x][2] == val)

        {

            return val;

        }

    }

    return 0;

}

int check\_Colwise(int ttt[3][3])

{

    int val;

    for (int x = 0; x < 3; x++)

    {

        val = ttt[0][x];

        if (ttt[1][x] == val && ttt[2][x] == val)

        {

            return val;

        }

    }

    return 0;

}

int check\_Diagonalwise(int ttt[3][3])

{

    int val;

    // primary diagonal

    val = ttt[0][0];

    if (ttt[1][1] == val && ttt[2][2] == val)

    {

        return val;

    }

    // secondary diagonal

    val = ttt[0][2];

    if (ttt[1][1] == val && ttt[2][0] == val)

    {

        return val;

    }

    return 0;

}

int main()

{

    int ttt[3][3];

    read\_Board(ttt);

    int emptycells = count\_EmptyCell(ttt);

    if (emptycells == 9)

    {

        printf("Empty");

    }

    else if (emptycells == 0)

    {

        int rowwin = check\_Rowwise(ttt);

        int colwin = check\_Colwise(ttt);

        int diagwin = check\_Diagonalwise(ttt);

        if (rowwin == 0 && colwin == 0 && diagwin == 0)

        {

            printf("Draw");

        }

        else if (rowwin != 0)

        {

            printf("Player%d wins", rowwin);

        }

        else if (colwin != 0)

        {

            printf("Player%d wins", colwin);

        }

        else if (diagwin != 0)

        {

            printf("Player%d wins", diagwin);

        }

    }

    else

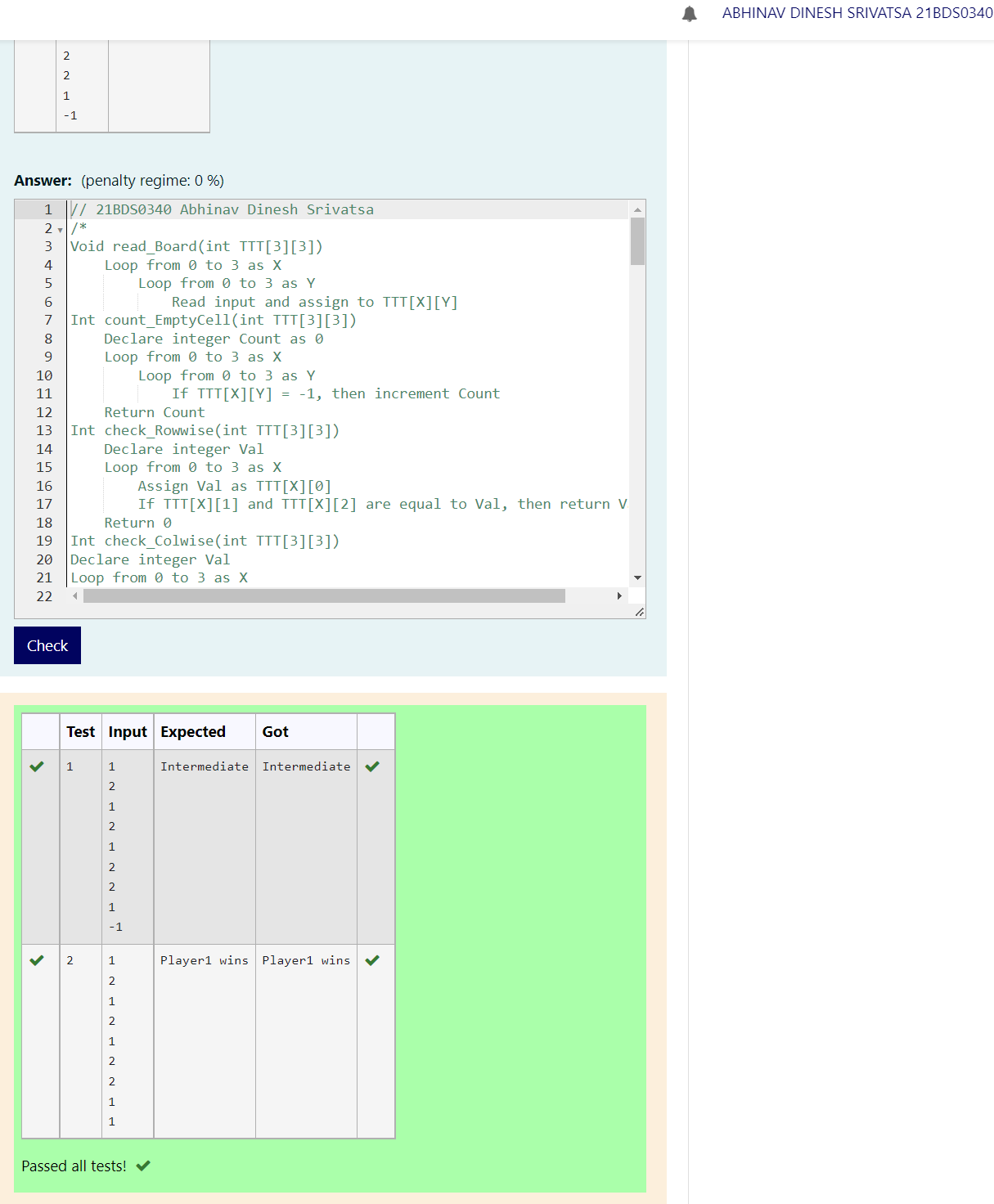
    {

        printf("Intermediate");

    }

}

### Output



## [Ex. No. M3\_CSQ1]

### AIM

Design a program in C to find the maximum of given three numbers using pointers.

### Algorithm / Pseudocode

Declare integer array Num with 3 spaces

Loop from 0 to 3 as X

Read input and assign to Num[X]

Declare integer Max as Num[0]

Declare pointer integer Store as Num

Loop from 0 to 3 as X

If Max < Store[X], then assign Max as Store[X]

Display Max

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int \*num = malloc(3 \* sizeof(int));

    for (int x = 0; x < 3; x++)

        scanf("%d", num + x);

    int max = \*num;

    for (int x = 0; x < 3; x++)

        if (max < \*(num + x))

            max = \*(num + x);

    free(num);

    printf("%d", max);

}

### Output

Graphical user interface, application

Description automatically generated

## [Ex. No. M3\_CSQ2]

### AIM

Create a program in C to dynamically allocate integer array. Display the elements of the array using dereferencing operator in the reverse order.

### Algorithm / Pseudocode

Declare integer N

Read input and assign to N

Declare integer pointer Arr and assign array of N spaces

Loop from 0 to N as X

Read input and assign to Arr[X]

Calculate Arr as Arr + N - 1

Loop from N - 1 to -1 as X

Display value of Arr

Decrement Arr

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int n;

    scanf("%d", &n);

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

    for (int x = 0; x < n; x++)

        scanf("%d", (arr + x));

    for (int x = n - 1; x > -1; x--)

        printf("%d\n", \*(arr + x));

    free(arr);

}

### Output

Graphical user interface

Description automatically generated

## [Ex. No. M3\_CSQ3]

### AIM

Consider one dimensional studentsMark array and find maximum and minimum scorer by defining a **function**. Use only pointers, not index to complete the task.

### Algorithm / Pseudocode

Int findMin(int \*)

Declare integer Min as Arr's value

Loop from 0 to 5 as X

If Min > Arr's value, then assign Min as Arr's value

Incerement Arr

Return Min

Int findMax(int \*)

Declare integer Max as Arr's value

Loop from 0 to 5 as X

If Max < Arr's value, then assign Max as Arr's value

Incerement Arr

Return Max

Declare intger pointer Arr and assign array of 5 spaces

Loop from 0 to 5 as X

Read input and assign to Arr

Increment Arr

Calculate Arr as Arr - 5 (Going back to root address)

Display min and max marks by calling findMin(Arr) and findMax(Arr)

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

int findMin\_21BDS0340(int \*arr)

{

    int min = \*arr;

    for (int x = 0; x < 5; x++)

        if (min > \*(arr + x))

            min = \*(arr + x);

    return min;

}

int findMax\_21BDS0340(int \*arr)

{

    int max = \*arr;

    for (int x = 0; x < 5; x++)

        if (max < \*(arr + x))

            max = \*(arr + x);

    return max;

}

int main()

{

    int \*arr = malloc(5 \* sizeof(int));

    for (int x = 0; x < 5; x++)

        scanf("%d", arr + x);

    printf("%d\n%d", findMax\_21BDS0340(arr), findMin\_21BDS0340(arr));

    free(arr);

}

### Output

Graphical user interface, application

Description automatically generated

## [Ex. No. M3\_CSQ4]

### AIM

The weather station of each city has the detail of rainfall in a year. Given the date and cm of rainfall recorded on that day, write a C program to determine the rainfall recorded in each month of the year and average monthly rainfall in the year.

### Algorithm / Pseudocode

Declare integer N

Read input and assign to N

Declare character array Date with 10 spaces

Declare integer array Rain with 12 spaces and assign all indices as 0

Declare integer Cm

Declare character array Month with 3 spaces

Loop from 0 to N

Read input and assign to Date

Read input and assign to Cm

Copy the month portion of Date and assign to Month

Calculate Rain[Month - 1] as itself + Cm

Declare integer Sum as 0

Loop from 0 to 12 as X

If Rain[X] is not 0, then display X + 1 and Rain[X]

Calculate Sum as Sum + Rain[X]

Display Sum / N

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

int main()

{

    int n;

    scanf("%d", &n);

    char date[10];

    int \*rain = calloc(12, sizeof(int));

    int cm;

    char month[3];

    for (int x = 0; x < n; x++)

    {

        scanf("%s", date);

        scanf("%d", &cm);

        strncpy(month, &date[3], 2);

        rain[atoi(month) - 1] += cm;

    }

    int sum = 0;

    for (int x = 0; x < 12; x++)

    {

        if (\*(rain + x) != 0)

        {

            printf("%d\n%d\n", (x + 1), \*(rain + x));

            sum += \*(rain + x);

        }

    }

    printf("%d", (int)round((float)sum / n));

    free(rain);

}

### Output

Graphical user interface

Description automatically generated with medium confidence

## [Ex. No. M3\_CSQ5]

### AIM

Design a C program to store Proctee's registered course details. Dynamically allocate memory for n students, scan number of courses, respective course code and credit, dynamically. Print the registration number, registered total credits of the proctees along with common courses.

### Algorithm / Pseudocode

Declare integer N, M and Temp

Read input and assign to N

Declare character array pointer Regno as N x 10

Declare integer array pointer Cred as N and assign values as 0

Declare integer Totalcourses as 0

Declare character array pointer Course as Totalcourses x 8

Loop from 0 to N as X

    Read input and store as Regno + X value

    Assign the last index of Regno + X as '\0'

    Read input and store as M

    Assign Course as reallocating Course by expanding the size to M + Totalcourses

    Loop from 0 to M as Y

        Read input and store as Course + Y + Totalcourses value

        Assign the last index of Course + Y + Totalcourses as '\0'

        Read input and assign to Temp

        Calculate Cred + X value as itself + Temp

    Calculate Totalcourses as Totalcourses + M

Loop from 0 to N as X

    Display Regno + X value and Cred + X value

Declare integer array pointer Count with Totalcourses amount of spaces and set all indiecs to 0

Loop from 0 to Totalcourses as X

    Loop from X + 1 to Totalcourses as Y

    If Course + X value is equal to Course + Y value, then increment Count + X value

Loop from 0 to Totalcourses as X

   If Count + X value is N - 1, then display Course + X value

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

    int n, m, temp;

    scanf("%d", &n);

    char(\*regno)[10] = malloc(n \* sizeof(\*regno));

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

    int total\_courses = 0;

    char(\*course)[8] = malloc(total\_courses \* sizeof(\*course));

    for (int x = 0; x < n; x++)

    {

        scanf("%s", \*(regno + x));

        \*(regno + x)[9] = '\0';

        scanf("%d", &m);

        course = realloc(course, (m + total\_courses) \* sizeof(\*course));

        for (int y = 0; y < m; y++)

        {

            scanf("%s", \*(course + y + total\_courses));

            \*(course + y + total\_courses)[7] = '\0';

            scanf("%d", &temp);

            \*(cred + x) += temp;

        }

        total\_courses += m;

    }

    for (int x = 0; x < n; x++)

        printf("%s\n%d\n", \*(regno + x), \*(cred + x));

    int \*count = calloc(total\_courses, sizeof(int));

    for (int x = 0; x < total\_courses; x++)

        for (int y = x + 1; y < total\_courses; y++)

            if (strcmp(\*(course + x), \*(course + y)) == 0)

                (\*(count + x))++;

    for (int x = 0; x < total\_courses; x++)

        if (\*(count + x) == (n - 1))

            printf("%s\n", \*(course + x));

    free(regno);

    free(cred);

    free(course);

    free(count);

}

### Output

Graphical user interface

Description automatically generated

## [Ex. No. M4\_CSQ1]

### AIM

Create a structure named student with two attributes, name and cgpa. Define structure variables s1, in the main block (local to main method), s2(global), two structure variable as an array. Scan over all four structure variables and display.

### Algorithm / Pseudocode

Declare structure student with 2 fields:

Character array Regis with 9 spaces

Float Cgpa

Declare student S2

Declare student S1 and array S with 2 spaces

Read input and store as S1's Regis and S1's Cgpa

Read input and store as S2's Regis and S2's Cgpa

Loop from 0 to 2 as X

Read input and store as S[x]'s Regis and S1[x]'s Cgpa

Display S1's Regis and S1's Cgpa

Display S2's Regis and S2's Cgpa

Loop from 0 to 2 as X

Display S[x]'s Regis and S1[x]'s Cgpa

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <string.h>

struct student

{

    char regis[9];

    float cgpa;

} s2;

int main()

{

    struct student s1, s[2];

    scanf("%s%f", s1.regis, &s1.cgpa);

    scanf("%s%f", s2.regis, &s2.cgpa);

    for (int x = 0; x < 2; x++)

    {

        scanf("%s%f", s[x].regis, &s[x].cgpa);

    }

    printf("%s\n%.1f\n", s1.regis, s1.cgpa);

    printf("%s\n%.1f\n", s2.regis, s2.cgpa);

    for (int x = 0; x < 2; x++)

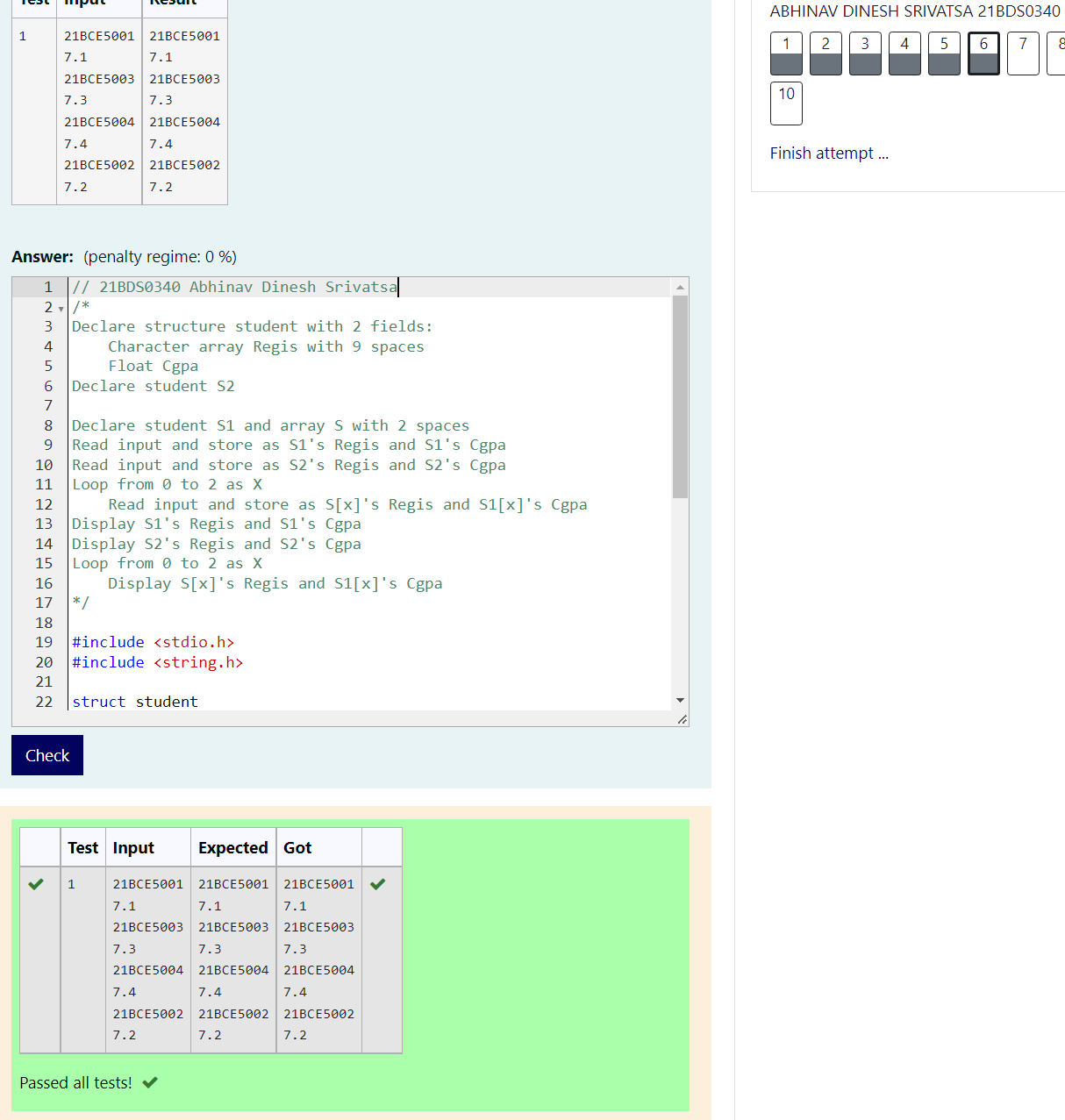
    {

        printf("%s\n%.1f\n", s[x].regis, s[x].cgpa);

    }

}

### Output



## [Ex. No. M4\_CSQ2]

### AIM

Create an employee structure with elements, empid, name, age, dept, designation, salary. Define array of employees and pass to a function to read the values and another function to sort the employees based on age attribute, display only empid.

### Algorithm / Pseudocode

Declare structure employee with 6 fields:

    Integer Empid

    Character array Name with 10 spaces

    Integer Age

    Character array Dept with 6 spaces

    Character array Desig with 15 spaces

    Integer Salary

void sort\_21BDS0340(struct employee \*E)

    Read inputs and store as E's Empid, Name, Age, Dept, Desig, Salary

void sort\_21BDS0340(struct employee E[3])

    Declare employee Temp

    Loop from 0 to 2 as X

    Loop from 0 to 2 - X as Y

        If E[Y]'s Age > E[Y+1]'s Age, then

                Assign Temp as E[Y]

                Assign E[Y] as E[Y+1]

                Assign E[Y+1] as Temp

Declare employee E with 3 spaces

Loop from 0 to 3 as X

    Call getValues\_21BDS0340 and pass E[X]

Call sort\_21BDS0340 and pass E

Loop from 2 to -1 as X

    Display E[X]'s Empid

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

struct employee

{

    int empid;

    char name[10];

    int age;

    char dept[6];

    char desig[15];

    int salary;

};

void getValues\_21BDS0340(struct employee \*e)

{

    scanf("%d%s%d%s%s%d", &e->empid, e->name, &e->age, e->dept, e->desig, &e->salary);

}

void sort\_21BDS0340(struct employee e[3])

{

    struct employee temp;

    for (int x = 0; x < 2; x++)

        for (int y = 0; y < 2 - x; y++)

            if (e[y].age > e[y + 1].age)

            {

                temp = e[y];

                e[y] = e[y + 1];

                e[y + 1] = temp;

            }

}

int main()

{

    struct employee e[3];

    for (int x = 0; x < 3; x++)

    {

        getValues\_21BDS0340(&e[x]);

    }

    sort\_21BDS0340(e);

    for (int x = 2; x >= 0; x--)

    {

        printf("%d\n", e[x].empid);

    }

}

### Output

Graphical user interface

Description automatically generated

## [Ex. No. M4\_CSQ3]

### AIM

The weather station of each city has the detail of rainfall in a year. Given the date and cm of rainfall recorded on that day, write a C program to determine the rainfall recorded in each month of the year and average monthly rainfall in the year

Note: please use structure for rainfall and date with necessary attributes. Please store date character array appropriately in date structure variable while returning from function.

### Algorithm / Pseudocode

Declare structure date with 3 fields:

Integer D

Integer M

Integer Y

Declare structure rain with 2 fields:

Date Date

Integer Cm

struct date dateToStruct(char Array[11])

Declare date D

Assign D.D as the date part of Array

Assign D.M as the month part of Array

Assign D.Y as the year part of Array

Return D

Declare integer N

Read input and assign to N

Declare rain array R with N spaces

Declare character array Datestr with 11 spaces

Loop from 0 to N as X

Read inputs and assign to Datestr and R[X]'s Cm

Assign R[X]'s Date by calling dateToStruct and passing Datestr

Declare integer array Month\_rain with 12 empty spaces

Declare integer Total\_rain as 0

Loop from 0 to N as X

Calculate Month\_rain[R[X]'s Date's D] as itself + R[X]'s Cm

Calculate Total\_rain as itself + R[X]'s Cm

Loop from 0 to N as X

If Month\_rain[X] is not 0, then display X + 1 and Month\_rain[X]

Display Total\_rain / N

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

struct date

{

    int d;

    int m;

    int y;

};

struct rain

{

    struct date date;

    int cm;

};

struct date dateToStruct(char array[11])

{

    struct date d;

    d.d = atoi(array);

    d.m = atoi(&array[3]);

    d.y = atoi(&array[6]);

    return d;

}

int main()

{

    int n;

    scanf("%d", &n);

    struct rain \*r = malloc(n \* sizeof(struct rain));

    char datestr[11];

    for (int x = 0; x < n; x++)

    {

        scanf("%s%d", datestr, &(r + x)->cm);

        (r + x)->date = dateToStruct(datestr);

    }

    int \*month\_rain = calloc(12, sizeof(int));

    int total\_rain = 0;

    for (int x = 0; x < n; x++)

    {

        month\_rain[(r + x)->date.m - 1] += (r + x)->cm;

        total\_rain += (r + x)->cm;

    }

    for (int x = 0; x < 12; x++)

        if (month\_rain[x] != 0)

            printf("%d\n%d\n", (x + 1), month\_rain[x]);

    printf("%d", (int)round((float)total\_rain / n));

    free(month\_rain);

    free(r);

}

### Output

Graphical user interface, application

Description automatically generated

## [Ex. No. M4\_CSQ4]

### AIM

Define C program with union definition named ID with three attributes, Aadhar, Pan, VoterId, based on choice from user, assign and access the element.

### Algorithm / Pseudocode

Declare union data with 3 fields:

Character array Aadhar with 13 spaces

Character array Pan with 11 spaces

Character array Voter with 11 spaces

Declare union data Id

Declare integer N

Read input and assign to N

Switch N

If 1, then read input and assign to Id's Aadhar and display the same

If 2, then read input and assign to Id's Pan and display the same

If 3, then read input and assign to Id's Voter and display the same

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <stdio.h>

union data

{

    char aadhar[13];

    char pan[11];

    char voter[11];

};

int main()

{

    union data id;

    int n;

    scanf("%d", &n);

    switch (n)

    {

    case 1:

        scanf("%s", id.aadhar);

        printf("%s", id.aadhar);

        break;

    case 2:

        scanf("%s", id.pan);

        printf("%s", id.pan);

        break;

    case 3:

        scanf("%s", id.voter);

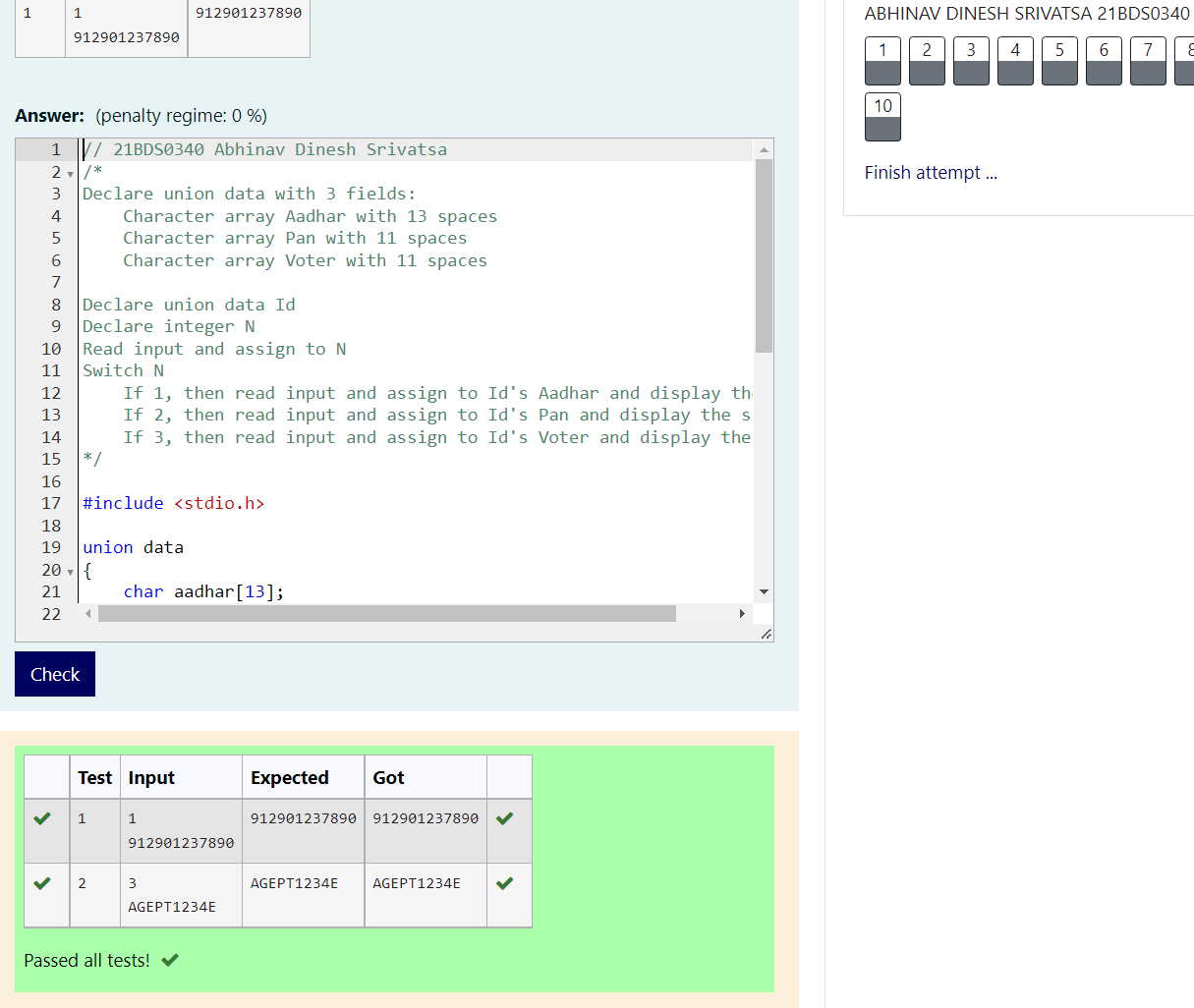
        printf("%s", id.voter);

        break;

    }

}

### Output



## [Ex. No. M4\_CSQ5]

### AIM

Details of employees (emp ID, name, joining date and mobile number) of a company is stored and maintained by the company’s IT department. On his birthday, the GM of the company wants to give a surprise gift of Rs.5000 for his employees who joined before 01/01/2010 and whose employee id is divisible by 5. Develop an algorithm and write a C program to display the name of the employees who are eligible to receive the gift and their mobile number

### Algorithm / Pseudocode

Declare typedef structure date with 3 fields:

Integer Day

Integer M

Integer Y

Declare typedef structure emp with 4 fields:

Integer Id

Character array Name with 15 spaces

Date Joindate

Character array Mobno with 11 spaces

date splitDate(char Str[10])

Declare Date D

Assign D's Day as the date part of Str

Assign D's Month as the month part of Str

Assign D's Year as the year part of Str

Return D

int isSurprised(emp E)

If E's Id is divisible by 5 and E's Joindate is less than 2010, then return 1

Return 0

Declare integer N

Read input and assign to N

Declare emp array E with N spaces

Declare character array Date with 11 spaces

Loop from 0 to N as X

Read 4 inputs and assign to (E + X)'s Id, (E + X)'s Name, Date and (E + X)'s Mobno

Assign (E + X)'s Joindate by calling splitDate and pass Date as the argument

Loop from 0 to N as X

If isSurprised of \*(E + X) is 1, then print (E + X)'s Name and Mobno

### Program Code

// 21BDS0340 ABhinav Dinesh Srivatsa

#include <stdio.h>

#include <stdlib.h>

typedef struct

{

    int day;

    int month;

    int year;

} date;

typedef struct

{

    int id;

    char name[15];

    date join\_date;

    char mob\_no[11];

} emp;

date splitDate(char str[11])

{

    date d;

    d.day = atoi(&str[0]);

    d.month = atoi(&str[3]);

    d.year = atoi(&str[6]);

    return d;

}

int isSurprised(emp e)

{

    if (e.id % 5 == 0 && e.join\_date.year < 2010)

        return 1;

    return 0;

}

int main()

{

    int n;

    scanf("%d", &n);

    emp \*e = malloc(n \* sizeof(emp));

    char date[11];

    for (int x = 0; x < n; x++)

    {

        scanf("%d%s%s%s", &(e + x)->id, (e + x)->name, date, (e + x)->mob\_no);

        (e + x)->join\_date = splitDate(date);

    }

    for (int x = 0; x < n; x++)

        if (isSurprised(\*(e + x)) == 1)

            printf("%s\n%s\n", (e + x)->name, (e + x)->mob\_no);

    free(e);

}

### Output

Graphical user interface

Description automatically generated

## [Ex. No. M5\_CSQ1]

### AIM

In an online examination system, each test will be scheduled for ‘x’ minutes. The student is free to take up the test on his convenience but once he starts the test, he must complete. Given the start time and the value of ‘x’ for an examination, develop an algorithm and write a ‘C++’ code for the examination system to calculate the finish time of the test

### Algorithm / Pseudocode

Test:

Declare integer Testtime

Declare integer Hour

Declare integer Minute

Public void readValues():

Declare integer TT, H, M

Assign Testtime as TT

Assign Hour as H

Assign Minute as M

Public void showEndTime():

Calculate Minute as Minute + Testtime

If Minute > 59, then calculate Hour as Hour + Minute / 60 and calculate minute as minute % 60

If Hour > 12, then calculate Hour as Hour - 12

If Hour < 10, then display "0" + Hour + ":"

Else display Hour + ":"

If Minute < 10, then display "0" + Minute

Else display Minute

Declare Test T

Call T.readValues()

Call T.showEndTIme()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Test

{

    int testtime;

    int hour;

    int minute;

public:

    void readValues()

    {

        int tt, h, m;

        cin >> h >> m >> tt;

        testtime = tt;

        hour = h;

        minute = m;

    }

    void showEndTime()

    {

        minute += testtime;

        if (minute > 59)

        {

            hour += minute / 60;

            minute %= 60;

        }

        if (hour > 12)

            hour -= 12;

        if (hour < 10)

            cout << "0" << hour << ":";

        else

            cout << hour << ":";

        if (minute < 10)

            cout << "0" << minute;

        else

            cout << minute;

    }

};

int main()

{

    Test t;

    t.readValues();

    t.showEndTime();

}

### Output

Graphical user interface, application

Description automatically generated

## [Ex. No. M5\_CSQ2]

### AIM

Define class rectangle with two attributes length and width along with member functions to setvalues and getvalues. Also define a member function to find the area of rectangle. At least create three objects and print the respective area for all three instances. Use this pointer in your methods.

### Algorithm / Pseudocode

Rectangle:

Declare integer Length

Declare integer Width

Public void setLength(int L):

Assign Length as L

Public void setWidth(int W):

Assign Width as W

Public int getLength():

Return Length

Public int getWidth():

Return Width

Public in area(int L, int W):

Return L \* W

Declare Rectangle array R with 3 spaces

Declare integer Length and Width

Loop from 0 to 3 as X

Read inputs and assign to Length and Width

Call R[X]'s setLength() and pass Length

Call R[X]'s setWidth() and pass Width

Loop from 0 to 3 as X

Call R[X]'s area() and pass R[X]'s getLength() and R[X]'s getWidth() and display

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Rectangle

{

    int length;

    int width;

public:

    void setLength(int length)

    {

        this->length = length;

    }

    void setWidth(int width)

    {

        this->width = width;

    }

    int getLength()

    {

        return this->length;

    }

    int getWidth()

    {

        return this->width;

    }

    int area(int length, int width)

    {

        return length \* width;

    }

};

int main()

{

    Rectangle r[3];

    int length, width;

    for (int x = 0; x < 3; x++)

    {

        cin >> length >> width;

        (r + x)->setLength(length);

        (r + x)->setWidth(width);

    }

    for (int x = 0; x < 3; x++)

    {

        cout << (r + x)->area((r + x)->getLength(), (r + x)->getWidth()) << "\n";

    }

}

### Output

Graphical user interface, treemap chart

Description automatically generated

## [Ex. No. M5\_CSQ3]

### AIM

Develop C++ program for a company to deal with employee information and create five objects. Find the employee whose designation is Manager and display respective emp id. Also create another method to display designation, by giving emp id.

### Algorithm / Pseudocode

Employee:

Declare public integer Id

Declare public character array Name with 10 spaces

Declare public character array Desig with 12 spaces

Void getId(Employee E[5], char Desig[10]):

Loop from 0 to 5 as X

If E[X]'s Desig == Desig, then display E[X]'s Id

Void getDesig(Employee E[5], int Id):

Loop from 0 to 5 as Y

If E[X]'s Id == Id, then display E[X]'s Desig

Declare Employee array E with 5 spaces

Declare integer Id

Declare character array Name with 10 spaces

Declare character array Desig with 12 spaces

Loop from 0 to 5 as X

Read inputs and assign to Id, Name and Desig

Assign E[X]'s Id as Id

Assign E[X]'s Name as Name

Assign E[X]'s Desig as Desig

Read input and assign to Desig

Read input and assign to Id

Call getId() and pass E and Desig

Call getDesig() and pass E and Id

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Employee

{

public:

    int id;

    char name[10];

    char desig[12];

};

void getID(Employee e[5], char desig[10])

{

    for (int x = 0; x < 5; x++)

        if (strcmp((e + x)->desig, desig) == 0)

            cout << (e + x)->id << "\n";

}

void getDesig(Employee e[5], int id)

{

    for (int x = 0; x < 5; x++)

        if ((e + x)->id == id)

            cout << (e + x)->desig << "\n";

}

int main()

{

    Employee e[5];

    int id;

    char name[10];

    char desig[12];

    for (int x = 0; x < 5; x++)

    {

        cin >> id >> name >> desig;

        (e + x)->id = id;

        strcpy((e + x)->name, name);

        strcpy((e + x)->desig, desig);

    }

    cin >> desig;

    cin >> id;

    getID(e, desig);

    getDesig(e, id);

}

### Output

Graphical user interface, chart, treemap chart

Description automatically generated

## [Ex. No. M5\_CSQ4]

### AIM

Create a class student with necessary attributes specific to each student along with constructors and define few instances. Define few common attributes common to all student with the help of static keyword and declare static member function. Display using a non-static member function, all students' regno.

### Algorithm / Pseudocode

Student:

Declare character array Regno with 10 spaces

Declare character array Name with 20 spaces

Declare integer Age

Declare public static character array Sec with 10 spaces

Declare public static character array Uni with 10 spaces

Student(char R[10], char N[20], int A):

Assign Regno as R

ASsign Name as N

Assign Age as A

Void dispRegno():

Display Regno

Declare character array Student's Sec with 10 spaces

Declare character array Student's Uni with 10 spaces

int main():

Read inputs and assign to Student's Sec and Student's Uni

Declare character array Regno with 2 x 10 spaces

Declare character array Name with 2 x 10 spaces

Declare integer array Age with 2 spaces

Loop from 0 to 2 as X

Read inputs and assign to Regno[X], Name[X] and Age[X]

Declare Student array S with 2 spaces and call constructor by passing Regno, Name and Age

Loop from 0 to 2 as X

Call S[X]'s dispRegno()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Student

{

    char regno[10];

    char name[20];

    int age;

public:

    static char sec[10];

    static char uni[10];

    Student(char regno[10], char name[20], int age)

    {

        strcpy(this->regno, regno);

        strcpy(this->name, name);

        this->age = age;

    }

    void dispRegno()

    {

        cout << this->regno << "\n";

    }

};

char Student::sec[10];

char Student::uni[10];

int main()

{

    cin >> Student::sec;

    cin >> Student::uni;

    char regno[2][10];

    char name[2][20];

    int age[2];

    for (int x = 0; x < 2; x++)

        cin >> regno[x] >> name[x] >> age[x];

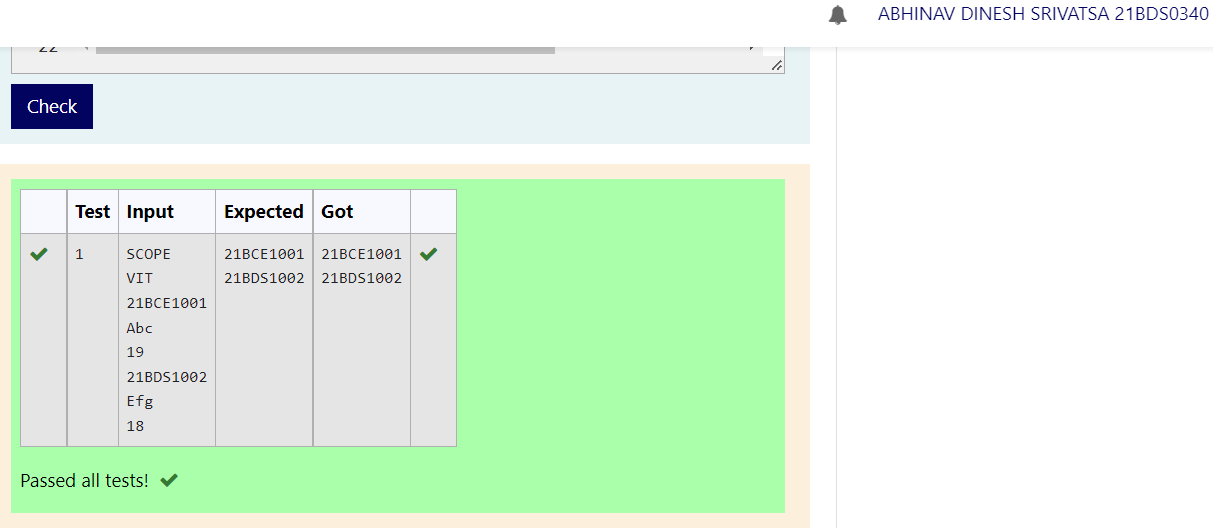
    Student s[2] = {Student(regno[0], name[0], age[0]), Student(regno[1], name[1], age[1])};

    for (int x = 0; x < 2; x++)

        s[x].dispRegno();

}

### Output



## [Ex. No. M5\_CSQ5]

### AIM

Create a class student with necessary attributes specific to each student. Define few instances, invoke setValues and getValues member functions. Display each student’s regno and phno through a member function, name displayInfo. Also create a non-member function to do the same task, name printInfo. [Function with object as an argument]

### Algorithm / Pseudocode

Student:

Declare character array Regno with 10 spaces

Declare character array Name with 15 spaces

Declare integer Age

Void setValues(char R[10], char N[15], int A):

Assign Regno as R

Assign Name as N

Assign Age as A

Char\* getRegno():

Return Regno

Char\* getName():

Return Name

Int getAge():

Return Age

Void displayInfo():

Call getRegno(), getName(), getAge() and display

Void printInfo(Student S):

Call S's getRegno(), S's getName(), S's getAge() and display

Int main():

Declare Student S1 and S2

Declare character arrays Regno and Name with 10 and 15 spaces respectively

Declare integer Age

Read inputs and assign to Regno, Name and Age

Call S1's setValues() and pass Regno, Name and Age as arguments

Read inputs and assign to Regno, Name and Age

Call S2's setValues() and pass Regno, Name and Age as arguments

Call S1's displayInfo()

Call printInfo() and pass S2 as argument

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Student

{

    char regno[10];

    char name[15];

    int age;

public:

    void setValues(char regno[10], char name[15], int age)

    {

        strcpy(this->regno, regno);

        strcpy(this->name, name);

        this->age = age;

    }

    char \*getRegno()

    {

        return this->regno;

    }

    char \*getName()

    {

        return this->name;

    }

    int getAge()

    {

        return this->age;

    }

    void displayInfo()

    {

        cout << this->getRegno() << "\n"

             << this->getName() << "\n"

             << this->getAge() << "\n";

    }

};

void printInfo(Student s)

{

    cout << s.getRegno() << "\n"

         << s.getName() << "\n"

         << s.getAge() << "\n";

}

int main()

{

    Student s1, s2;

    char regno[10], name[15];

    int age;

    cin >> regno >> name >> age;

    s1.setValues(regno, name, age);

    cin >> regno >> name >> age;

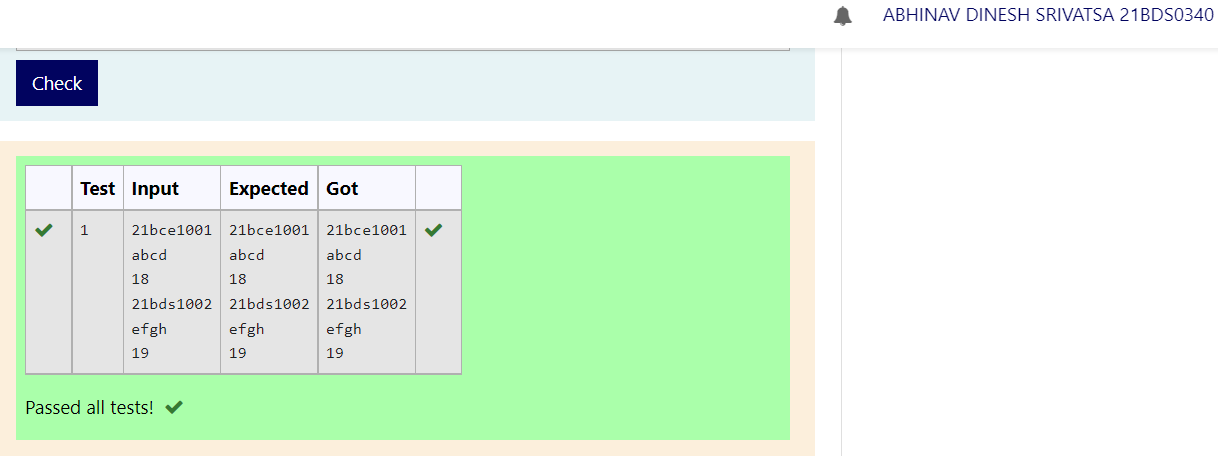
    s2.setValues(regno, name, age);

    s1.displayInfo();

    printInfo(s2);

}

### Output



## [Ex. No. M5\_CSQ6]

### AIM

Define an inline function in [CPP](https://moovit.vit.ac.in/mod/vpl/view.php?id=31822) program for calculating factorial of n

### Algorithm / Pseudocode

Int factorial(int N):

If N = 0, then return 1

Return N \* factorial(N - 1)

Int main():

Declare integer N

Read input and assign to N

Call factorial() and pass N as argument and display

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

inline int factorial(int n)

{

    if (n == 0)

        return 1;

    return n \* factorial(n - 1);

}

int main()

{

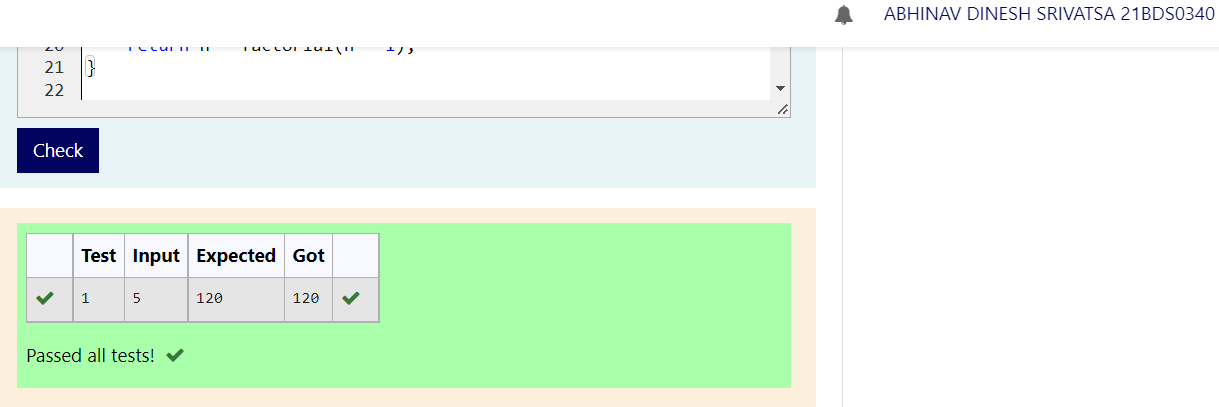
    int n;

    cin >> n;

    cout << factorial(n);

}

### Output



## [Ex. No. M5\_CSQ7]

### AIM

Develop [cpp](https://moovit.vit.ac.in/mod/vpl/view.php?id=31822) function to swap two values

### Algorithm / Pseudocode

Void swap(int \*X, int \*Y):

Declare integer pointer Temp and assign \*X

Assign \*X as \*Y

Assign \*Y as \*Temp

Int main():

Declare integers X and Y

Read inputs and assign to X and Y

Call swap() and pass &X and &Y as arguments

Display X and Y

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

void swap(int \*x, int \*y)

{

    int temp = \*x;

    \*x = \*y;

    \*y = temp;

}

int main()

{

    int x, y;

    cin >> x >> y;

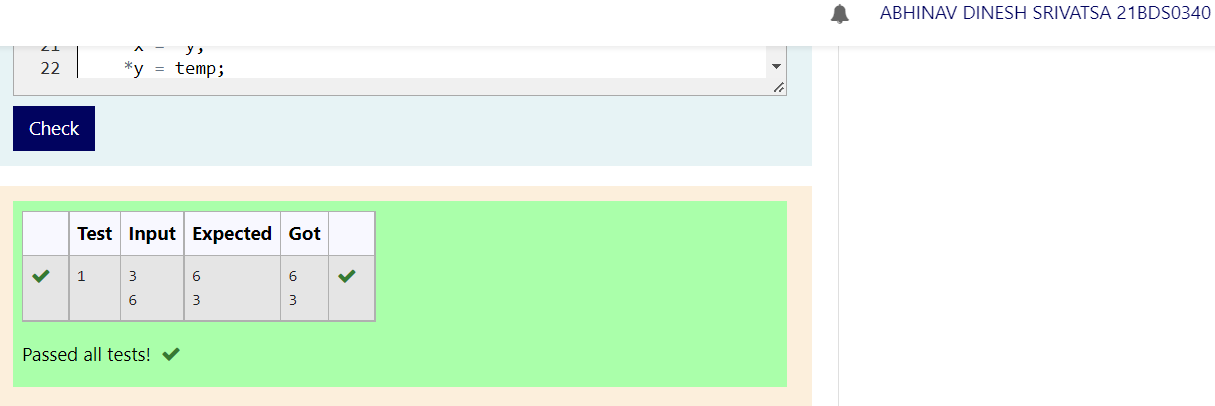
    swap(&x, &y);

    cout << x << "\n"

         << y << "\n";

}

### Output



## [Ex. No. M5\_CSQ8]

### AIM

Define [cpp](https://moovit.vit.ac.in/mod/vpl/view.php?id=31822) function to print -, 10 times if no argument passed, other print the character that is being passed along with times argument. If choice is 1, invoke print function with no argument, otherwise pass the arguments received as input.

### Algorithm / Pseudocode

Void dispChar(int N = 10, Char C = '-'):

Loop from 0 to N as X

Display C

Int main():

Declare integer Choice

Read input and assign to Choice

If Choice = 1, then call dispChar()

Else

Declare character C

Declare integer N

Read inputs and assign to C and N

Call dispChar() and pass N and C as arguments

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

void dispChar(int n = 10, char c = '-')

{

    for (int x = 0; x < n; x++)

        cout << c;

}

int main()

{

    int choice;

    cin >> choice;

    if (choice == 1)

        dispChar();

    else

    {

        char c;

        int n;

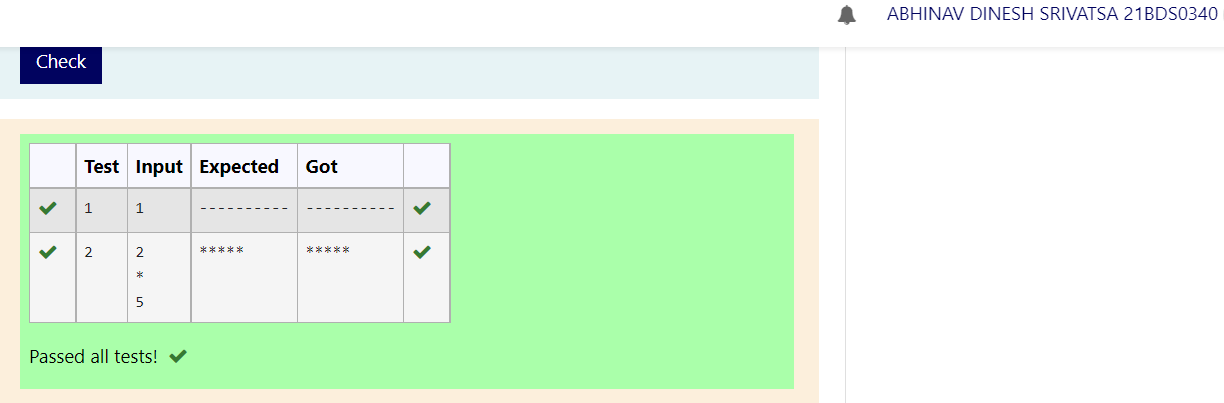
        cin >> c >> n;

        dispChar(n, c);

    }

}

### Output



## [Ex. No. M5\_CSQ9]

### AIM

Develop a friend function to calculate total fare for a ticket. There are ‘n’ passengers in the ticket and they are from a family of a railway employee. Discount is given for their travel based on the cader of the employee.

If Cader A - 10%, B - 15%, C - 20%

### Algorithm / Pseudocode

Fare:

Declare integer Passengers

Declare character array From with 20 spaces

Declare character array To with 20 spaces

Declare integer Ticket\_cost

Declare character Cader

Fare(int P, char F[20], char T[20], int Tc, char C):

Assign Passengers as P

Assign From as F

Assign To as T

Assign Ticket\_cost as Tc

Assign Cader as C

Declare Price as friend class

Price:

Int calculate(Fare F):

Switch F's Cader

If 'A', then return F's Ticket\_cost \* F's Passengers \* 0.9

If 'B', then return F's Ticket\_cost \* F's Passengers \* 0.85

If 'C', then return F's Ticket\_cost \* F's Passengers \* 0.8

Return 0

Int main():

Declare integers P and Tc

Declare character C and arrays Fr and T with 20 spaces

Read input and assign to P, Fr, T, Tc and C respectively

Declare Fare F() passing P, Fr, T, Tc and C as arguments

Declare Price Pr

Call Pr's calculate() and pass F and display

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Fare

{

    int passengers;

    char from[20];

    char to[20];

    int ticket\_cost;

    char cader;

public:

    Fare(int p, char f[20], char t[20], int tc, char c)

    {

        this->passengers = p;

        strcpy(this->from, f);

        strcpy(this->to, t);

        this->ticket\_cost = tc;

        this->cader = c;

    }

    friend class Price;

};

class Price

{

public:

    int calculate(Fare f)

    {

        switch (f.cader)

        {

        case 'A':

            return f.ticket\_cost \* f.passengers \* 0.9;

        case 'B':

            return f.ticket\_cost \* f.passengers \* 0.85;

        case 'C':

            return f.ticket\_cost \* f.passengers \* 0.8;

        }

        return 0;

    }

};

int main()

{

    int p, tc;

    char fr[20], t[20], c;

    cin >> p >> fr >> t >> tc >> c;

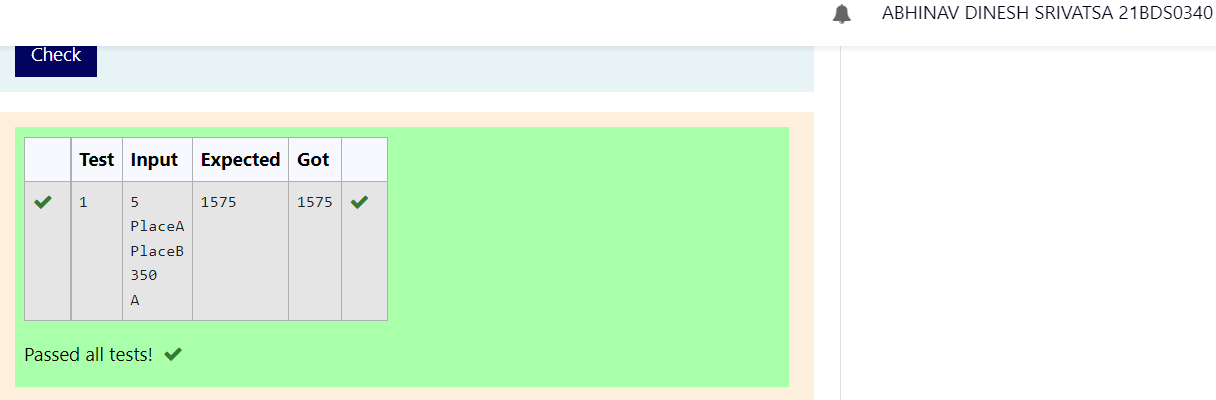
    Fare f(p, fr, t, tc, c);

    Price pr;

    cout << pr.calculate(f);

}

### Output



## [Ex. No. M5\_CSQ10]

### AIM

Develop a [cpp](https://moovit.vit.ac.in/mod/vpl/view.php?id=31822) program with a class named train with attributes trainNumber, array of coach objects, static data member named NumberOfCoaches along with member functions getTrain, printTrainInfo, addCoach, removeCoach.  Define a coach class with data members coachNumber[Eg: A1,B1..], coachClass[Eg: First, Second, Sleeper..], Ac/NonAc, capacity along with getCoach and printCoachInfo member functions.

Note: Declare class train as friend in class coach.

In main method, create a train instance and add two coaches and print NumberOfCoaches static value and total train capacity

### Algorithm / Pseudocode

Coach:

Declare character array CoachNumber with 3 spaces

Declare character array CoachClass with 20 spaces

Declare boolean Ac

Declare integer Capacity

Void getCoach():

Declare charcater array IsAC with 6 spaces

Read inputs and assign to CoachNumber, CoachClass, IsAC and Capacity

If IsAC is "True", then assign Ac as true

Else, assign false

Void printCoachInfo():

Declare character array IsAc with 4 spaces

If Ac is true, then assign IsAc as ""

Else assign "Not"

Display CoachNumber, CoachClass, IsAc "Ac" and Capacity

Declare friend class Train

Train:

Declare integer TrainNumber;

Declare Coach array Coaches with 10 spaces

Declare static integer NumberOfCoaches

Void getTrain():

Read input and assign to TrainNumber

Void PrintTrainInfo():

Display TrainNumber and NumberOfCoaches

Void addCoach():

Call Coaches[NumberOfCoaches]'s getCoach()

Increment NumberOfCoaches

Int getCapcaity():

Declare integer Capacity as 0

Loop from 0 to NumberOfCoaches

Add Coaches[x]'s Capacity to Capacity

Return Capacity

Declare integer Train's NumberOfCoaches as 0

Int main():

Declare Train T

Call T's getTrain()

Call T's addCoach() twice

Display T's NumberOfCoaches and T's getCapacity()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Coach

{

    char coachNumber[3];

    char coachClass[20];

    bool ac;

    int capacity;

    void getCoach()

    {

        char isAC[6];

        cin >> this->coachNumber >> this->coachClass >> isAC >> this->capacity;

        if (strcmp(isAC, "True"))

            this->ac = true;

        else

            this->ac = false;

    }

    void printCoachInfo()

    {

        char isAC[4];

        if (this->ac)

            strcpy(isAC, "");

        else

            strcpy(isAC, "Not");

        cout << this->coachNumber << "\n" << this->coachClass << "\n" << isAC << " AC\n" << this->capacity << "\n";

    }

    friend class Train;

};

class Train

{

    int trainNumber;

    Coach coaches[10];

public:

    static int NumberOfCoaches;

    void getTrain()

    {

        cin >> this->trainNumber;

    }

    void printTrainInfo()

    {

        cout << this->trainNumber << "\n" << NumberOfCoaches << "\n";

    }

    void addCoach()

    {

        coaches[NumberOfCoaches].getCoach();

        NumberOfCoaches++;

    }

    int getCapacity()

    {

        int capacity = 0;

        for (int x = 0; x < NumberOfCoaches; x++)

            capacity += this->coaches[x].capacity;

        return capacity;

    }

};

int Train::NumberOfCoaches = 0;

int main()

{

    Train t;

    t.getTrain();

    t.addCoach();

    t.addCoach();

    cout << t.NumberOfCoaches << "\n"

         << t.getCapacity();

}

### Output

Graphical user interface

Description automatically generated

### [Ex. No. M6\_CSQ1]

### AIM

Create a class named person with attributes name, age, phno with getPerson and printPerson member functions. Extend the class named employee with additional attributes empId, designation along with getEmp and displayEmp member functions. Define an array of employee instances and invoke the respective employee member functions and display only empId of all employees in ascending order.

### Algorithm / Pseudocode

Person:

Declare character array Name with 20 spaces

Declare integer Age

Declare character array Phno with 11 spaces

Void getPerson():

Read inputs and assign to Name, Age and Phno

Void printPerson():

Display Name, Age and Phno

Employee, extends Person:

Declare character array Desig with 20 spaces

Declare integer EmpId

Void getEmp():

Call Person's getPerson()

Read input and assign to EmpId and Desig

Void dispEmp():

Call Person's printPerson()

Display EmpId and Desig

Int main():

Declare integer N

Read input and assign as N

Declare Employee array E with N spaces

Loop from 0 to N as X

Call E[X]'s getEmp()

Loop from 0 to N as X

Loop from X to N as Y

If E[X]'s EmpId > E[Y]'s EmpId, then swap them

Loop from 0 to N as X

Display E[X]'s EmpId

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Person

{

    char name[20];

    int age;

    char phno[11];

public:

    void getPerson()

    {

        cin >> this->name >> this->age >> this->phno;

    }

    void printPerson()

    {

        cout << this->name << "\n"

             << this->age << "\n"

             << this->phno << "\n";

    }

};

class Employee : private Person

{

    char desig[20];

public:

    int empId;

    void getEmp()

    {

        Person::getPerson();

        cin >> this->empId >> this->desig;

    }

    void dispEmp()

    {

        Person::printPerson();

        cout << this->empId << "\n"

             << this->desig << "\n";

    }

};

int main()

{

    int n;

    cin >> n;

    Employee e[n];

    for (int x = 0; x < n; x++)

        e[x].getEmp();

    for (int x = 0; x < n; x++)

        for (int y = x + 1; y < n; y++)

            if (e[x].empId > e[y].empId)

            {

                Employee temp = e[y];

                e[y] = e[x];

                e[x] = temp;

            }

    for (int x = 0; x < n; x++)

        cout << e[x].empId << "\n";

}

### Output

Chart, treemap chart

Description automatically generated

### [Ex. No. M6\_CSQ2]

### AIM

Create a class named person with attributes name, age, phno with getPerson and printPerson member functions. Extend the class named employee with additional attributes empId, designation along with getEmp and displayEmp member functions. Further extend it to PartTimeEmployee class with NumberofHoursWorked and HourlyWage attributes, member functions getPartTimeEmployeeDetail and displayPartTimeEmployeeDetail. Define a PartTimeEmployee instance, invoke the respective member functions and display the details.

### Algorithm / Pseudocode

Person:

Declare character array Name with 20 spaces

Declare integer Age

Declare character array Phno with 11 spaces

Void getPerson():

Read inputs and assign to Name, Age and Phno

Void printPerson():

Display Name, Age and Phno

Employee, extends Person:

Declare character array Desig with 20 spaces

Declare integer EmpId

Void getEmp():

Call Person's getPerson()

Read input and assign to EmpId and Desig

Void dispEmp():

Call Person's printPerson()

Display EmpId and Desig

PartTimeEmployee, extends Employee:

Declare integer NumberofHoursWorked

Declare integer HourlyWage

Void getPartTimeEmployee():

Call Employee's getEmp()

Read inputs and assign to NumberofHoursWorked and HourlyWage

Void displayPartTimeEmployeeDetail():

Call Employee's dispEmp()

Display NumberofHoursWorked and HourlyWage

Int main():

Declare PartTimeEmployee P

Call P's getPartTimeEmployee()

Call P's displayPartTimeEmployeeDetail()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Person

{

    char name[20];

    int age;

    char phno[11];

public:

    void getPerson()

    {

        cin >> this->name >> this->age >> this->phno;

    }

    void printPerson()

    {

        cout << this->name << "\n"

             << this->age << "\n"

             << this->phno << "\n";

    }

};

class Employee : private Person

{

    char desig[20];

public:

    int empId;

    void getEmp()

    {

        Person::getPerson();

        cin >> this->empId >> this->desig;

    }

    void dispEmp()

    {

        Person::printPerson();

        cout << this->empId << "\n"

             << this->desig << "\n";

    }

};

class PartTimeEmployee : private Employee

{

    int NumberofHoursWorked;

    int HourlyWage;

public:

    void getPartTimeEmployee()

    {

        Employee::getEmp();

        cin >> NumberofHoursWorked >> HourlyWage;

    }

    void displayPartTimeEmployeeDetail()

    {

        Employee::dispEmp();

        cout << NumberofHoursWorked << "\n"

             << HourlyWage << "\n";

    }

};

int main()

{

    PartTimeEmployee p;

    p.getPartTimeEmployee();

    p.displayPartTimeEmployeeDetail();

}

### Output

Graphical user interface

Description automatically generated

### [Ex. No. M6\_CSQ3]

### AIM

Create a class named ResearchAssitant with two parent classes Student[Regno , name, cgpa, phno, school  and get, print member functions] and Research[ResearchTitle, GuideName and get, print member functions], with new attributes NumberOfPapersPublished, NumberOfConfernceAttended with get and print member functions. Define an array ResearchAssitant instances and find the one who has published more papers.

### Algorithm / Pseudocode

Student:

Declare character array Regno with 10 spaces

Declare character array Name with 20 spaces

Declare float Cgpa

Declare character array Phno with 11 spaces

Declare character array School with 20 spaces

Void getStudent():

Read inputs and assign as Regno, Name, Cgpa, Phno and School

Void printStudent():

Display Regno, Name, Cgpa, Phno and School

Research:

Declare character array ResearchTitle with 20 spaces

Declare character array GuideName with 20 spaces

Void getResearch():

Read inputs and assign to ResearchTitle and GuideName

Void printResearch():

Display ResearchTitle and GuideName

ResearchStudent:

Declare integer NumberOfPapersPublished

Declare integer NumberOfConferencesAttended

Void getResearchStudent():

Call Student's getStudent()

Call Research's getResearch()

Read inputs and assign to NumberOfPapersPublished and NumberOfConferencesAttended

Void printResearchStudent():

Call Student's printStudent()

Call Research's printResearch()

Display NumberOfPapersPublished and NumberOfConferencesAttended

Int main():

Declare integer N

Read input and assign to N

Declare ResearchStudent array Rs with N spaces

Loop from 0 to N as X

Call Rs[X]'s getResearchStudent()

Declare integer Max as Rs[0]'s NumberOfPapersPublished

Declare character array Regno with 10 spaces

Assign Regno as Rs[0]'s Regno

Loop from 1 to N as X

If Rs[X]'s NumberOfPapersPublished > Max, then

Assign Max as Rs[X]'s NumberOfPapersPublished

Assign Regno as Rs[X]'s Regno

Display Regno

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Student

{

public:

    char regno[10];

    char name[20];

    float cgpa;

    char phno[11];

    char school[20];

    void getStudent()

    {

        cin >> regno >> name >> cgpa >> phno >> school;

    }

    void printStudent()

    {

        cout << regno << "\n"

             << name << "\n"

             << cgpa << "\n"

             << phno << "\n"

             << school << "\n";

    }

};

class Research

{

    char ResearchTitle[20];

    char GuideName[20];

public:

    void getResearch()

    {

        cin >> ResearchTitle >> GuideName;

    }

    void printResearch()

    {

        cout << ResearchTitle << "\n"

             << GuideName << "\n";

    }

};

class ResearchStudent : public Student, public Research

{

public:

    int NumberOfPapersPublished;

    int NumberOfConferencesAttended;

public:

    void getResearchStudent()

    {

        Student::getStudent();

        Research::getResearch();

        cin >> NumberOfPapersPublished >> NumberOfConferencesAttended;

    }

    void printResearchStudent()

    {

        Student::printStudent();

        Research::printResearch();

        cout << NumberOfPapersPublished << "\n"

             << NumberOfConferencesAttended << "\n";

    }

};

int main()

{

    int n;

    cin >> n;

    ResearchStudent rs[n];

    for (int x = 0; x < n; x++)

        rs[x].getResearchStudent();

    int max = rs[0].NumberOfPapersPublished;

    char regno[10];

    strcpy(regno, rs[0].regno);

    for (int x = 1; x < n; x++)

        if (rs[x].NumberOfPapersPublished > max)

        {

            max = rs[x].NumberOfPapersPublished;

            strcpy(regno, rs[x].regno);

        }

    cout << regno;

}

### Output

Chart

Description automatically generated

### [Ex. No. M6\_CSQ4]

### AIM

Hybrid\_Inheritance

Define a class named person[name, age, getPerson, displayPerson], extend the class and define a Student class[regno, cgpa, getStudent, displayStudent].

Create a class named GateScore[GateRegistrationNumber, Score, ExamCode, getGate, displayGate].

Develop a class PG\_Student[deptName, getPGS, displayPGS] inherited from two parents, Student class and GateScore class.

In main method, create n instance of PG students and invoke all member functions. Find the student and display VIT Regno and GATE regno, who has scored highest Gate Score.

### Algorithm / Pseudocode

Person:

Declare character array Name with 20 spaces

Declare integer Age

Void getPerson():

Read inputs and assign to Name and Age

Void displayPerson():

Display Name and Age

Student, extends Person:

Declare character array Regno with 10 spaces

Declare float Cgpa

Void getStudent():

Call getPerson()

Read inputs and assign to Regno and Cgpa

Void displayStudent():

Call displayPerson()

Display Regno and Cgpa

Char\* getRegno():

Return Regno

GateScore:

Declare character array GateRegistrationNumber with 20 spaces

Declare integer Score

Declare character array ExamCode with 20 spaces

Void getGate():

Read inputs and assign to GateRegistrationNumber, Score and ExamCode

Void displayGate():

Display GateRegistrationNumber, Score and ExamCode

Char\* getGateRegno():

Return GateRegistrationNumber

Int getGateScore():

Return Score

PG\_Student, extends Student and GateScore:

Declare character array DeptName with 10 spaces

Void getPGS():

Call getStudent()

Call getGate()

Read input and assign to DeptName

Void displayPGS():

Call displayStudent()

Call displayGate()

Display DeptName

Int main():

Declare integer N

Read input and assign to N

Declare PG\_Student array Pgs with N spaces

Loop from 0 to N as X

Call Pgs[X]'s getPGS()

Declare integer Max as Pgs[0]'s getGateScore()

Declare character arrays Regno and GateRegno with 10 and 20 spaces

Assign Regno as Pgs[0]'s getRegno()

Assign GateRegno as Pgs[0]'s getGateRegno()

Loop from 0 to N as X

If Pgs[X]'s getGateScore() > Max, then

Assign Max as Pgs[X]'s getGateScore()

Assign Regno as Pgs[X]'s getRegno()

Assign GateRegno as Pgs[X]'s getGateRegno()

Display Regno and GateRegno

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Person

{

    char name[20];

    int age;

public:

    void getPerson()

    {

        cin >> name >> age;

    }

    void displayPerson()

    {

        cout << name << "\n"

             << age << "\n";

    }

};

class Student : public Person

{

    char regno[10];

    float cgpa;

public:

    void getStudent()

    {

        getPerson();

        cin >> regno >> cgpa;

    }

    void displayStudent()

    {

        displayPerson();

        cout << regno << "\n"

             << cgpa << "\n";

    }

    char \*getRegno()

    {

        return this->regno;

    }

};

class GateScore

{

    char GateRegistrationNumber[20];

    int Score;

    char ExamCode[20];

public:

    void getGate()

    {

        cin >> GateRegistrationNumber >> Score >> ExamCode;

    }

    void displayGate()

    {

        cout << GateRegistrationNumber << "\n"

             << Score << "\n"

             << ExamCode << "\n";

    }

    char \*getGateRegno()

    {

        return this->GateRegistrationNumber;

    }

    int getGateScore()

    {

        return this->Score;

    }

};

class PG\_Student : public Student, public GateScore

{

    char deptName[10];

public:

    void getPGS()

    {

        getStudent();

        getGate();

        cin >> deptName;

    }

    void displayPgs()

    {

        displayStudent();

        displayGate();

        cout << deptName << "\n";

    }

};

int main()

{

    int n;

    cin >> n;

    PG\_Student pgs[n];

    for (int x = 0; x < n; x++)

        pgs[x].getPGS();

    int max = pgs[0].getGateScore();

    char regno[10], gateRegno[20];

    strcpy(regno, pgs[0].getRegno());

    strcpy(gateRegno, pgs[0].getGateRegno());

    for (int x = 0; x < n; x++)

        if (pgs[x].getGateScore() > max)

        {

            max = pgs[x].getGateScore();

            strcpy(regno, pgs[x].getRegno());

            strcpy(gateRegno, pgs[x].getGateRegno());

        }

    cout << regno << "\n"

         << gateRegno << "\n";

}

### Output

Chart, bar chart

Description automatically generated

### [Ex. No. M6\_CSQ5]

### AIM

Hierarchical\_Inheritance:

Define a class called employee[empid, name and phno]. Create two subclasses named permanent employee[ department, basicSalary, HRA[x% of basicSalary], DA[y% of basicSalary] and temporary employee[contractor name, consolidated salary]. Please create Read and Display member functions in each class. Develop a calculateSalary constant member function in permanent employee class.

Get a choice from user,

if entered 1, create permanent employee instance, scan the values and print the empid and total salary [basicSalary+HRA+DA],

if entered 2, display empid and consolidated salary of temporary employee instance

Note: space provided in input, accordingly read the values

### Algorithm / Pseudocode

Employee:

Declare character array Empid with 10 spaces

Declare character array Name with 20 spaces

Declare character array Phno with 11 spaces

Void readEmployee():

Read inputs and assign to Empid, Name and Phno

Void displayEmployee():

Display Empid, Name and Phno

Char\* getEmpid():

Return Empid

PermanentEmployee, extends Employee:

Declare character array Department with 15 spaces

Declare integers BasicSalary, HRA and DA

Void readPermanentEmployee():

Read inputs and assign to Department, BasicSalary, HRA and DA

Void displayPermanentEmployee():

Display Department, BasicSalary, HRA and DA

Int calcSalary():

Return BasicSalary + (HRA + DA) \* BasicSalary / 100

TemporaryEmployee, extends Employee:

Declare character array ContractorName with 20 spaces

Declare integer Salary

Void readTemporaryEmployee():

Read inputs and assign to ContractorName and Salary

Void displayTemporaryEmployee():

Display ContractorName and Salary

Int getSalary():

Return Salary

Int main():

Declare integer Choice

Read input and assign to Choice

Switch Choice,

If 1, then

Declare PermanentEmployee Pe

Call Pe's readPermanentEmployee()

Display Pe's getEmpid() and calcSalary()

If 2, then

Declare TemporaryEmployee Te

Call Te's readTemporaryEmployee()

Display Te's getEmpid() and getSalary()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Employee

{

    char empid[10];

    char name[20];

    char phno[11];

public:

    void readEmployee()

    {

        cin >> empid;

        cin.ignore();

        cin.getline(name, 20);

        cin >> phno;

    }

    void displayEmployee()

    {

        cout << empid << "\n"

             << name << "\n"

             << phno << "\n";

    }

    char \*getEmpid()

    {

        return this->empid;

    }

};

class PermanentEmployee : public Employee

{

    char department[15];

    int basicSalary;

    int HRA;

    int DA;

public:

    void readPermanentEmployee()

    {

        readEmployee();

        cin.ignore();

        cin.getline(department, 15);

        cin >> basicSalary >> HRA >> DA;

    }

    void displayPermanentEmployee()

    {

        displayEmployee();

        cout << department << "\n"

             << basicSalary << "\n"

             << HRA << "\n"

             << DA << "\n";

    }

    int calcSalary()

    {

        return basicSalary + (HRA + DA) \* basicSalary / 100;

    }

};

class TemporaryEmployee : public Employee

{

    char contractorName[20];

    int salary;

public:

    void readTemporaryEmployee()

    {

        readEmployee();

        cin.ignore();

        cin.getline(contractorName, 20);

        cin >> salary;

    }

    void displayTemporaryEmployee()

    {

        displayEmployee();

        cout << contractorName << "\n"

             << salary << "\n";

    }

    int getSalary()

    {

        return this->salary;

    }

};

int main()

{

    int choice;

    cin >> choice;

    switch (choice)

    {

    case 1:

        PermanentEmployee pe;

        pe.readPermanentEmployee();

        cout << pe.getEmpid() << "\n"

             << pe.calcSalary() << "\n";

        break;

    case 2:

        TemporaryEmployee te;

        te.readTemporaryEmployee();

        cout << te.getEmpid() << "\n"

             << te.getSalary() << "\n";

    }

}

### Output

Graphical user interface, treemap chart

Description automatically generated

### [Ex. No. M6\_CSQ6]

### AIM

Create a bank customer class.

Let the customer class have name, account number, account type, status (active/dormant), balance, and last transaction date as data members. Define n customer instances.

Write a C++ program to do the following, collect account number and choice

choice = 1 : Display the name of the customer for the collected account number

2 :  Withdraw funds given the amount. Deny the withdrawal transaction if balance becomes less than 1000.

3 :  Credit funds given the amount.

For each customer, change the status of the account from active to dormant if no transactions occurred in the past one year.

Display all customers, account number, balance, and status.

### Algorithm / Pseudocode

Customer:

Declare character array Name with 20 spaces

Declare character array Acc\_num with 25 spaces

Declare character array Acc\_type with 10 spaces

Declare character array Status with 8 spaces

Declare integer Balance

Declare character array Last\_date with 11 spaces

Void getInput():

Read inputs and assign to Name, Acc\_num, Acc\_type, Status, Balance and Last\_date

Char\* getAccNum():

Return Acc\_num

Char\* getName():

Return Name

Int getBalance():

Return Balance

Char\* getStatus():

Return Status

Void withdraw(int Amt):

If Balance - Amt < 1000, then return

Calculate Balance as Balance - Amt

Void deposit(int Amt):

Calculate Balance as Balance + Amt

Void checkStatus():

Declare Date as "19-05-2022" (current date)

Declare Day as the day of Date

Declare Month as the month of Date

Declare Year as the year of Date

Declate Last\_day as the day of Last\_date

Declare Last\_month as the month of Last\_date

Declare Last\_year as the year of Last\_date

If Year > Last\_year

If Month > Last\_month, then assign Status as "dormant"

If Month = Last\_month and Day > Last\_day, then assign status as "dormant"

Int main():

Declare integer N

Read input and assign as N

Declare Customer array C with N spaces

Loop from 0 to N as X

Call C[X]'s getInput()

Declare character array Acc\_num with 25 spaces

Declare integer Choice, Amt and Index

Read inputs and assign to Acc\_num and Choice

Loop from 0 to N as X

If Acc\_num = C[X]'s getAccNum(), then assign Index as X

Switch Choice

If 1

Display C[Index]'s getName()

If 2

Read input and assign to Amt

Call C[Index]'s withdraw() and pass Amt

If 3

Read input and assign to Amt

Call C[Index]'s deposit() and pass Amt

Loop from 0 to N as X

Call C[X]'s checkStatus()

Display C[X]'s getAccNum(), C[X]'s getBalance(), C[X]'s getStatus()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Customer

{

    char name[20];

    char acc\_num[25];

    char acc\_type[10];

    char status[8];

    int balance;

    char last\_date[11];

public:

    void getInput()

    {

        cin >> this->name >> this->acc\_num >> this->acc\_type >> this->status >> this->balance >> this->last\_date;

    }

    char \*getAccNum()

    {

        return this->acc\_num;

    }

    char \*getName()

    {

        return this->name;

    }

    int getBalance()

    {

        return this->balance;

    }

    char \*getStatus()

    {

        return this->status;

    }

    void withdraw(int amt)

    {

        if (this->balance - amt < 1000)

            return; // transaction denied

        this->balance -= amt;

    }

    void deposit(int amt)

    {

        this->balance += amt;

    }

    void checkStatus()

    {

        char \*date = (char \*)"19-05-2022";

        int day = atoi(date);

        int month = atoi(&date[3]);

        int year = atoi(&date[6]);

        int last\_day = atoi(this->last\_date);

        int last\_month = atoi(&this->last\_date[3]);

        int last\_year = atoi(&this->last\_date[6]);

        if (year > last\_year)

        {

            if (month > last\_month)

                strcpy(this->status, "dormant");

            if (month == last\_month && day > last\_day)

                strcpy(this->status, "dormant");

        }

    }

};

int main()

{

    int n;

    cin >> n;

    Customer c[n];

    for (int x = 0; x < n; x++)

        c[x].getInput();

    char acc\_num[25];

    int choice, amt, index;

    cin >> acc\_num >> choice;

    for (int x = 0; x < n; x++)

        if (strcmp(c[x].getAccNum(), acc\_num) == 0)

            index = x;

    switch (choice)

    {

    case 1:

        cout << c[index].getName();

        break;

    case 2:

        cin >> amt;

        c[index].withdraw(amt);

        break;

    case 3:

        cin >> amt;

        c[index].deposit(amt);

    }

    for (int x = 0; x < n; x++)

    {

        c[x].checkStatus();

        cout << c[x].getAccNum() << "\n"

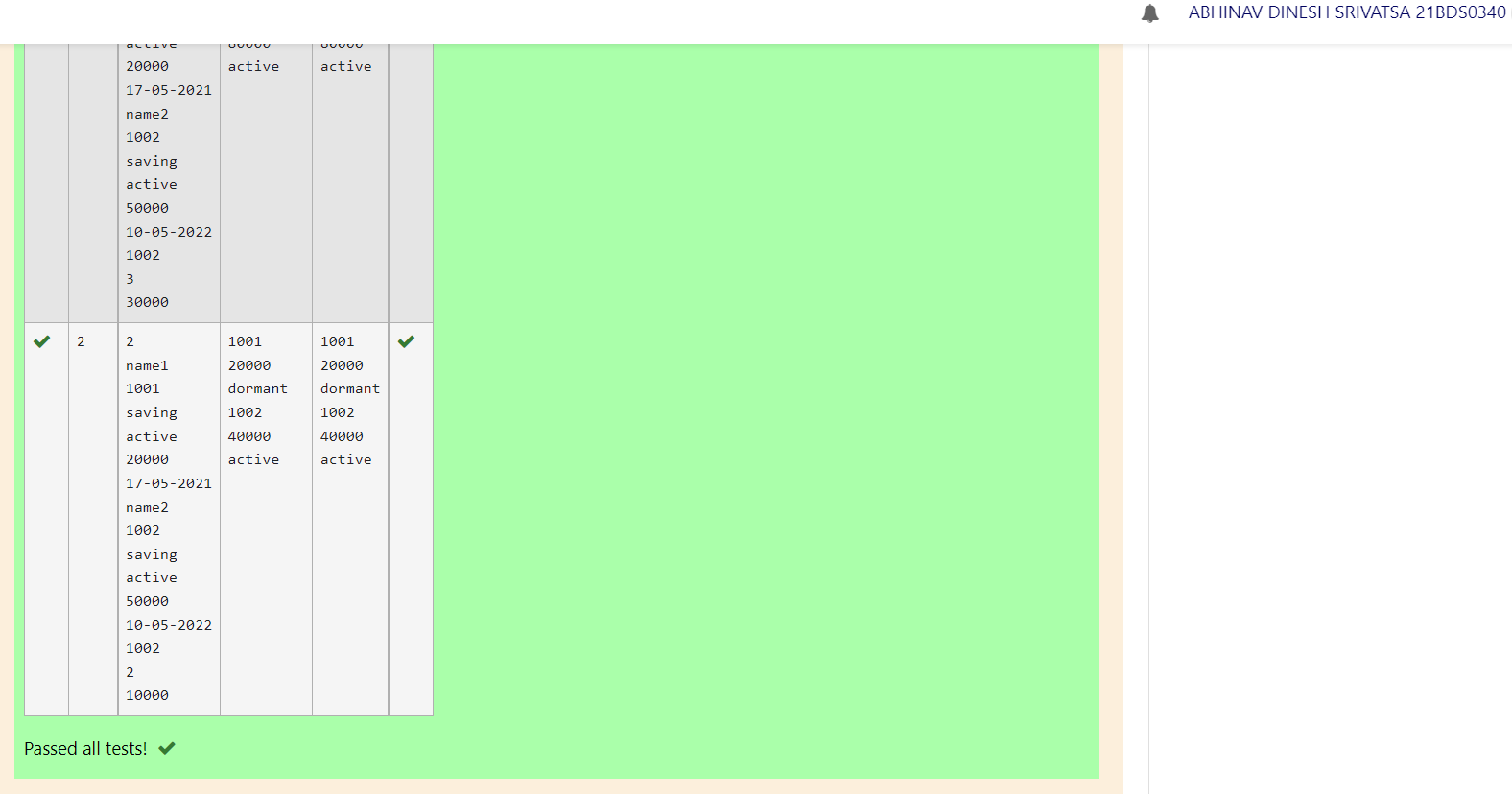
             << c[x].getBalance() << "\n"

             << c[x].getStatus() << "\n";

    }

}

### Output



### [Ex. No. M6\_CSQ7]

### AIM

Create a class called Person with the following data members (Name, Age, and Gender).

Create a class called student with the additional details such as (Reg.No, School, CGPA).

Create another class called Sport with level (State/National) as the data member.

Add 0.5 point to the student’s CGPA if he is a State level player and 1 point to the student’s CGPA if he is a national level player.

Write a C++ program to print the name and registration number of the student along with his CGPA using inheritance.

### Algorithm / Pseudocode

Person:

Declare character array Name with 15 spaces

Declare integer Age

Declare character Gender

Char\* getName():

Return Name

Sport:

Declare character array Level with 10 spaces

Char\* getLevel():

Return Level

Student, extends Person and Sport:

Declare character array Regno with 10 spaces

Declare character array School with 10 spaces

Declare float Cgpa

Void getInput():

Read inputs and assign to Name, Age, Gender, Regno, School, Cgpa, Level

Float getCGPA():

Return Cgpa

Void setCGPA(float C):

Assign Cgpa as C

Char\* getRegno():

Return Regno

Int main():

Declare Student S

Call S's getInput()

If S's getLevel() = "State", then call S's setCGPA and pass S's getCGPA + 0.5

Else if S's getLevel() = "National", then call S's setCGPA and pass S's getCGPA + 1.0

Display S's getName(), S's getRegno() and S's getCGPA()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

#include <iomanip>

using namespace std;

class Person

{

protected:

    char name[15];

    int age;

    char gender;

public:

    char \*getName()

    {

        return this->name;

    }

};

class Sport

{

protected:

    char level[10];

public:

    char \*getLevel()

    {

        return this->level;

    }

};

class Student : public Person, public Sport

{

    char regno[10];

    char school[10];

    float cgpa;

public:

    void getInput()

    {

        cin >> this->name >> this->age >> this->gender >> this->regno >> this->school >> this->cgpa >> this->level;

    }

    float getCGPA()

    {

        return this->cgpa;

    }

    void setCGPA(float cgpa)

    {

        this->cgpa = cgpa;

    }

    char \*getRegno()

    {

        return this->regno;

    }

};

int main()

{

    Student s;

    s.getInput();

    if (strcmp(s.getLevel(), "State") == 0)

        s.setCGPA(s.getCGPA() + 0.5f);

    else if (strcmp(s.getLevel(), "National") == 0)

        s.setCGPA(s.getCGPA() + 1.0f);

    cout << fixed;

    cout << setprecision(1);

    cout << s.getName() << "\n"

         << s.getRegno() << "\n"

         << s.getCGPA();

}

### Output

Graphical user interface

Description automatically generated

### [Ex. No. M6\_CSQ8]

### AIM

Create a class called Person with the following data members (Name, Age, and Gender) along with default and parametrized constructors.

Extend the above class as student with the additional details such as (Reg.No, School, CGPA) along with default and parametrized constructors.

Define an instance for child and invoke parametrized constructors for both classes. Display name and regno.

### Algorithm / Pseudocode

Person:

Declare character array Name with 15 spaces

Declare integer Age

Declare character Gender

Person(char N[15], int A, char G):

Assign Name as N

Assign Age as A

Assing Gender as G

Char\* getName():

Return Name

Student, extends Person:

Declare character array Regno with 10 spaces

Declare character array School with 10 spaces

Declare float Cgpa

Student(char N[15], int A, char G, char R[10], char S[10], float C):Person(N, A, G):

Assign Regno as R

Assign School as S

Assign Cgpa as C

Char\* getRegno():

Return Regno

Int main():

Declare character array Name with 15 spaces

Declare integer Age

Declare character Gender

Declare character array Regno with 10 spaces

Declare character array School with 10 spaces

Declare float Cgpa

Read inputs and assign to Name, Age, Gender, Regno, School and Cgpa

Declare Student S and pass Name, Age, Gender, Regno, School and Cgpa

Display S's getName() and S's getRegno()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Person

{

    char name[15];

    int age;

    char gender;

public:

    Person(char name[15], int age, char gender)

    {

        strcpy(this->name, name);

        this->age = age;

        this->gender = gender;

    }

    char \*getName()

    {

        return this->name;

    }

};

class Student : public Person

{

    char regno[10];

    char school[10];

    float cgpa;

public:

    Student(char name[15], int age, char gender, char regno[10], char school[10], float cgpa) : Person(name, age, gender)

    {

        strcpy(this->regno, regno);

        strcpy(this->school, school);

        this->cgpa = cgpa;

    }

    char \*getRegno()

    {

        return this->regno;

    }

};

int main()

{

    char name[15];

    int age;

    char gender;

    char regno[10];

    char school[10];

    float cgpa;

    cin >> name >> age >> gender >> regno >> school >> cgpa;

    Student s(name, age, gender, regno, school, cgpa);

    cout << s.getName() << "\n"

         << s.getRegno();

}

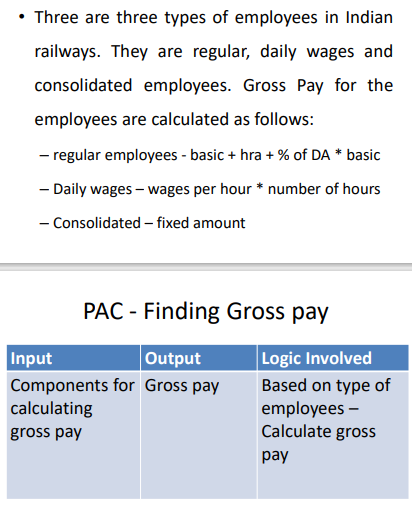
### Output

Graphical user interface

Description automatically generated

### [Ex. No. M7\_CSQ1]

### AIM



### Algorithm / Pseudocode

Regular:

Declare integer Basic

Declare integer Hra

Declare integer Da

Regular(int B, int H, int D):

Assign Basic as B

Assign Hra as H

Assign Da as D

Int getPay():

Return Basic + Hra + Da \* Basic / 100

Daily:

Declare integer Wpa

Declare integer Hours

Daily(int W, int H):

Assign Wpa as W

Assign Hours as H

Int getPay():

Return Wpa \* Hours

Consolidated:

Declare integer Amt

Consolidated(int A):

Assign Amt as A

Int getPay():

Return Amt

Int main():

Declare integers Choice, I1, I2, I3

Read input and assign to Choice

Switch Choice

If 1, then read inputs and assign to I1, I2, I3

Declare Regular R() and pass I1, I2, I3

Display R's getPay()

If 2, then read inputs and assign to I1, I2

Declare Daily D() and pass I1, I2

Display D's getPay()

If 3, then read input and assign to I1

Declare Consolidated C() and pass I1

Display C's getPay()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Regular

{

    int basic;

    int hra;

    int da;

public:

    Regular(int basic, int hra, int da)

    {

        this->basic = basic;

        this->hra = hra;

        this->da = da;

    }

    int getPay()

    {

        return basic + hra + da \* basic / 100;

    }

};

class Daily

{

    int wpa;

    int hours;

public:

    Daily(int wpa, int hours)

    {

        this->wpa = wpa;

        this->hours = hours;

    }

    int getPay()

    {

        return wpa \* hours;

    }

};

class Consolidated

{

    int amount;

public:

    Consolidated(int amount)

    {

        this->amount = amount;

    }

    int getPay()

    {

        return amount;

    }

};

int main()

{

    int choice, i1, i2, i3;

    cin >> choice;

    switch (choice)

    {

    case 1:

    {

        cin >> i1 >> i2 >> i3;

        Regular r(i1, i2, i3);

        cout << r.getPay();

        break;

    }

    case 2:

    {

        cin >> i1 >> i2;

        Daily d(i1, i2);

        cout << d.getPay();

        break;

    }

    case 3:

    {

        cin >> i1;

        Consolidated c(i1);

        cout << c.getPay();

    }

    }

}

### Output

Shape, rectangle

Description automatically generated

### [Ex. No. M7\_CSQ2]

### AIM

Create a class called complex with real and imaginary part as data members. Define setter and getter functions.

Kindly Overload

binary - and +as member function and non member function respectively

binary \* as non member function

unary operator  - as member function

### Algorithm / Pseudocode

Complex:

Declare integer Real

Declare integer Imag

Void display():

Display Real, Imag

Void setReal(int R):

Assign Real as R

Int getReal():

Return Real

Void setImag(int I):

Assign Imag as I

Int getImag():

Return Imag

Complex operator - (Complex C):

Declare Complex Temp

Assign Temp's Real as Real - C's Real

Assign Temp's Imag as Imag - C's Imag

Return Temp

Complex operator - ():

Declare Complex Temp

Assign Temp's Real as -Real

Assign Temp's Imag as -Imag

Return Temp

Declare Complex operator+(Complex C1, Complex C2) as friend

Declare Complex operator\*(Complex C1, Complex C2) as friend

Complex operator+(Complex C1, Complex C2):

Declare Complex Temp

Assign Temp's Real as C1's Real + C2's Real

Assign Temp's Imag as C1's Imag + C2's Imag

Return Temp

Complex operator\*(Complex C1, Complex C2):

Declare Complex Temp

Assign Temp's Real as C1's Real \* C2's Real - C1's Imag \* C2's Imag

Assign Temp's Imag as C1's Real \* C2's Imag + C1's Imag \* C2's Real

Return Temp

Int main():

Declare integer N, I

Read input and assign to N

Declare Complex array C with N spaces

Loop from 0 to N as X

Read input and assign to I

Call C[x]'s setReal() and pass I

Read input and assign to I

Call C[x]'s setImag() and pass I

Declare character Op

Read input and assign to Op

Declare Complex Out and assign C[0]

If N = 1 and Op = '-', then calculate Out as -C[0]

Else loop from 1 to N as X

Switch Op

If '+', then calculate Out as Out + C[X]

If '-', then calculate Out as Out - C[X]

If '\*', then calculate Out as Out \* C[X]

Call Out's display()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Complex

{

    int real;

    int imag;

public:

    void display()

    {

        cout << real << "\n"

             << imag;

    }

    void setReal(int real)

    {

        this->real = real;

    }

    int getReal()

    {

        return real;

    }

    void setImag(int imag)

    {

        this->imag = imag;

    }

    int getImag()

    {

        return imag;

    }

    Complex operator-(Complex c)

    {

        Complex temp;

        temp.real = this->real - c.real;

        temp.imag = this->imag - c.imag;

        return temp;

    }

    Complex operator-()

    {

        Complex temp;

        temp.real = -this->real;

        temp.imag = -this->imag;

        return temp;

    }

    friend Complex operator+(Complex c1, Complex c2);

    friend Complex operator\*(Complex c1, Complex c2);

};

Complex operator+(Complex c1, Complex c2)

{

    Complex temp;

    temp.real = c1.real + c2.real;

    temp.imag = c1.imag + c2.imag;

    return temp;

}

Complex operator\*(Complex c1, Complex c2)

{

    Complex temp;

    temp.real = c1.real \* c2.real - c1.imag \* c2.imag;

    temp.imag = c1.real \* c2.imag + c1.imag \* c2.real;

    return temp;

}

int main()

{

    int n, i;

    cin >> n;

    Complex c[n];

    for (int x = 0; x < n; x++)

    {

        cin >> i;

        c[x].setReal(i);

        cin >> i;

        c[x].setImag(i);

    }

    char op;

    cin >> op;

    Complex out = c[0];

    if (n == 1 && op == '-')

        out = -c[0];

    else

        for (int x = 1; x < n; x++)

        {

            switch (op)

            {

            case '+':

                out = out + c[x];

                break;

            case '-':

                out = out - c[x];

                break;

            case '\*':

                out = out \* c[x];

                break;

            }

        }

    out.display();

}

### Output

Shape, square

Description automatically generated

### [Ex. No. M7\_CSQ3]

### AIM

Define a height class with data members feet and inches. Overload the following relational operators <, <=, >, >=

### Algorithm / Pseudocode

Height:

Declare integer Feet

Declare integer Inch

Void setHeight(int F, int I):

Assign Feet as F

Assign Inch as I

Bool operator < (Height H):

If Feet < H's Feet, return true

Else if Feet = H's Feet and Inch < H's Inch, then return true

Return false

Bool operator <= (Height H):

If Feet < H's Feet, return true

Else if Feet = H's Feet and Inch <= H's Inch, then return true

Return false

Bool operator > (Height H):

If Feet > H's Feet, return true

Else if Feet = H's Feet and Inch > H's Inch, then return true

Return false

Bool operator >= (Height H):

If Feet > H's Feet, return true

Else if Feet = H's Feet and Inch >= H's Inch, then return true

Return false

Int main():

Declare Height H1, H2

Declare integers I1, I2, J1, J2

Read inputs and assign to I1, J1, I2, J2

Call H1's setHeight() and pass I1, J1

Call H2's setHeight() and pass I2, J2

If H1 > H2, then display "Taller"

Else if H1 >= H2, then display "Same"

Else if H1 < H2, then display "Shorter"

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

using namespace std;

class Height

{

    int feet;

    int inch;

public:

    void setHeight(int feet, int inch)

    {

        this->feet = feet;

        this->inch = inch;

    }

    bool operator<(Height h)

    {

        if (this->feet < h.feet)

            return true;

        else if (this->feet == h.feet)

            if (this->inch < h.inch)

                return true;

        return false;

    }

    bool operator<=(Height h)

    {

        if (this->feet < h.feet)

            return true;

        else if (this->feet == h.feet)

            if (this->inch <= h.inch)

                return true;

        return false;

    }

    bool operator>(Height h)

    {

        if (this->feet > h.feet)

            return true;

        else if (this->feet == h.feet)

            if (this->inch > h.inch)

                return true;

        return false;

    }

    bool operator>=(Height h)

    {

        if (this->feet > h.feet)

            return true;

        else if (this->feet == h.feet)

            if (this->inch >= h.inch)

                return true;

        return false;

    }

};

int main()

{

    Height h1;

    Height h2;

    int i1, j1, i2, j2;

    cin >> i1 >> j1 >> i2 >> j2;

    h1.setHeight(i1, j1);

    h2.setHeight(i2, j2);

    if (h1 > h2)

        cout << "Taller";

    else if (h1 >= h2)

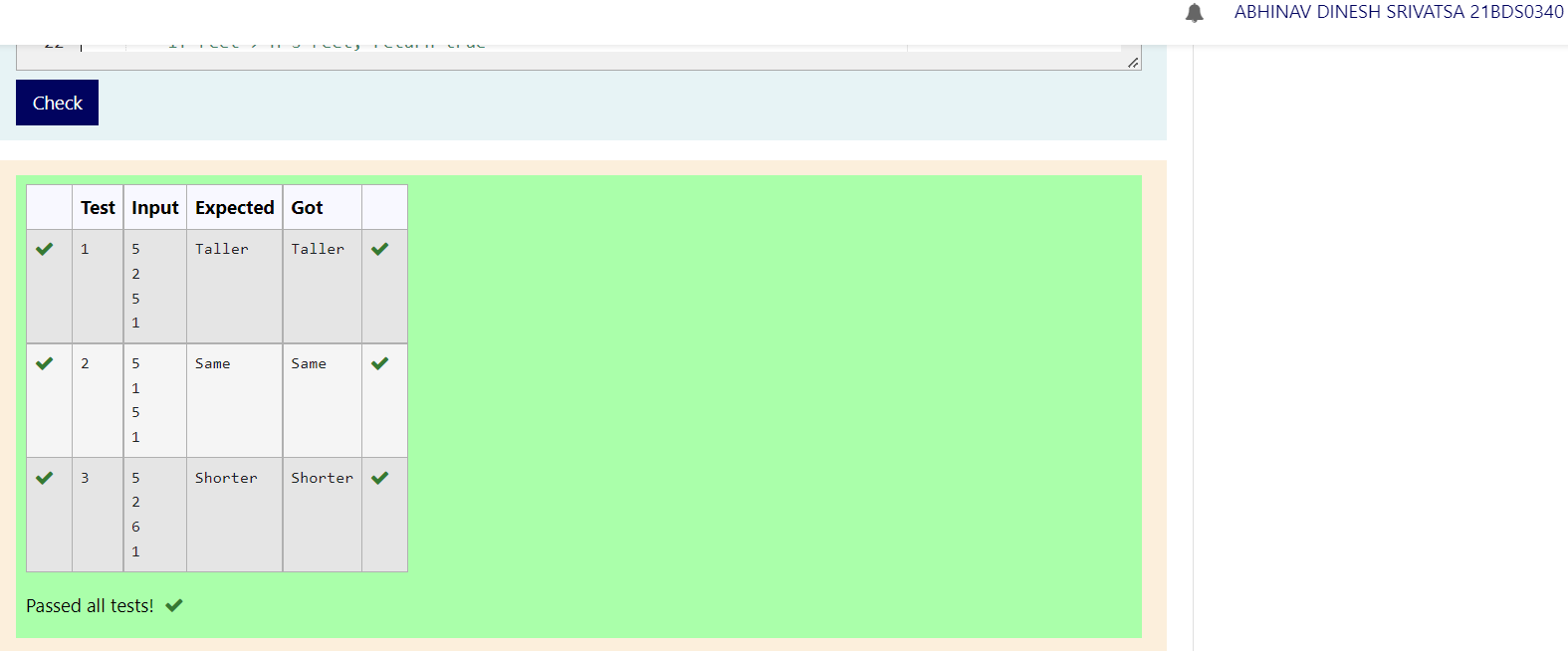
        cout << "Same";

    else if (h1 < h2)

        cout << "Shorter";

}

### Output



### [Ex. No. M7\_CSQ4]

### AIM

Define a class called student and define an array of instances, say 3. Print an instance (based on input) using exertion operator. In case of accessing beyond array size, warn the user by overloading [] operator.

### Algorithm / Pseudocode

Student:

Declare character array Name with 10 spaces

Declare character array Regno with 10 spaces

Void setData(char N[10], char R[10]):

Assign Name as N

Assign Regno as R

Declare ostream& operator<<(ostream &Out, Student &S) as friend function

Ostream& operator<<(ostream &Out, Student &S):

Display S's Name and S's Regno

Return Out

StudArr:

Declare Student array S with 4 spaces

Student& operator[](int I):

If I is between 1 and 3, then return S[I]

Display "Out of bound"

Call S[0]'s setData() and pass "" and ""

Return S[0]

Int main():

Declare StudArr St

Declare character arrays Data1 and Data2 with 10 spaces

Loop from 1 to 4 as X

Read input and assign to Data1 and Data2

Call St[X]'s setData() and pass Data1 and Data2

Declare integer N

Read input and assign to N

Display St[N]

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Student

{

    char name[10];

    char regno[10];

public:

    void setData(char name[10], char regno[10])

    {

        strcpy(this->name, name);

        strcpy(this->regno, regno);

    }

    friend ostream &operator<<(ostream &out, Student &s);

};

ostream &operator<<(ostream &out, Student &s)

{

    out << s.name << " " << s.regno;

    return out;

}

class StudArr

{

public:

    Student s[4];

    Student &operator[](int i)

    {

        if (0 < i && i < 4)

            return s[i];

        cout << "Out of bound";

        char n[10] = "", r[10] = "";

        s[0].setData(n, r);

        return s[0];

    }

};

int main()

{

    StudArr st;

    char data1[10], data2[10];

    for (int x = 1; x < 4; x++)

    {

        cin >> data1 >> data2;

        st[x].setData(data1, data2);

    }

    int n;

    cin >> n;

    cout << st[n];

}

### Output

Graphical user interface, treemap chart

Description automatically generated with medium confidence

### [Ex. No. M7\_CSQ5]

### AIM

Define a parent class person with data members name, phno with necessary member functions. Extend the class to Faculty [empid, TwoCurrentSemesterCourses - coursecode] and Staff[ staff if and department]. Create virtual functions read and display methods in parent class. Based on choice create either Faculty or Staff instance.

Note: Use parent class reference to hold child object

### Algorithm / Pseudocode

Parent:

Declare character arrays Name and Phno

Create pure virtual void read() and display()

Faculty, extends Parent:

Declare character arrays Empid, Course1 and Course2

Void read():

Read inputs and assign to Name, Phno, Empid, Course1 and Course2

Void display():

Display Course1 and Course2

Staff, extends Parent:

Declare character arrays Staffid and Dep

Void read():

Read inputs and assign to Name, Phno, Staffid and Dep

Void display():

Display Dep

Int main():

Declare Parent pointer P

Declare integer Choice

Read input and assign as Choice

Switch Choice

If 1, then declare Faculty F and assign it's address to P

If 2, then declare Staff S and assign it's address to P

Call P's read() and display()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Parent

{

protected:

    char name[15];

    char phno[11];

public:

    virtual void read() = 0;

    virtual void display() = 0;

};

class Faculty : public Parent

{

    char empid[10];

    char course1[10];

    char course2[10];

public:

    void read()

    {

        cin.ignore();

        cin.getline(name, 15);

        cin >> phno >> empid >> course1 >> course2;

    }

    void display()

    {

        cout << course1 << "\n"

             << course2;

    }

};

class Staff : public Parent

{

    char staffid[10];

    char dep[15];

public:

    void read()

    {

        cin.ignore();

        cin.getline(name, 15);

        cin >> phno >> staffid >> dep;

    }

    void display()

    {

        cout << dep;

    }

};

int main()

{

    Parent \*p;

    int choice;

    cin >> choice;

    switch (choice)

    {

    case 1:

    {

        Faculty f;

        p = &f;

        break;

    }

    case 2:

    {

        Staff s;

        p = &s;

    }

    }

    p->read();

    p->display();

}

### Output

Graphical user interface

Description automatically generated

### [Ex. No. M7\_CSQ6]

### AIM

A number of jobs may be submitted to an Operating System (OS).

Assume that the OS follows shortest job first scheduling that is when a number of jobs namely j1, j2, j3 are submitted at the same time, the job which requires minimum time to complete is served first.

Similar to case when three customers enter a shop at the same time, customer who is getting least number of items is served first. An OS named as OS1\_X supports two types of scheduling

• Based on time required to complete job

• Based on memory required to complete job Design an OOP model and implement the same in C++ to perform the scheduling.

Assume that all the jobs are submitted at the same time. Give the solution, based on the choice of scheduling opted by the user.

### Algorithm / Pseudocode

Job:

Declare integers Id, Time and Space

Declare character Name

Void readData():

Read inputs and assign to Id, Name, Time and Space

Int getTime():

Return Time

Int getSpace():

Return Space

Int getId():

Return Id

OS:

Void sortTime(int N, Job \*J):

Declare Job Temp

Loop from 0 to N as X

Loop from X + 1 to N as Y

If J[X]'s getTime() > J[X]'s getTime(), then swap J[X] and J[Y]

Void sortSpace(int N, Job \*J):

Declare Job Temp

Loop from 0 to N as X

Loop from X + 1 to N as Y

If J[X]'s getSpace() > J[X]'s getSpace(), then swap J[X] and J[Y]

Int main():

Declare integer N

Declare character array Sort

Read inputs and assign to N and Sort

Declare Job array J with N spaces

Loop from 0 to N as X

Call J[X]'s readData()

Declare OS OS1\_X

If Sort = "Time", then call OS1\_x's sortTime() and pass N and J

Else if Sort = "Space", then call OS1\_x's sortTime() and pass N and J

Loop from 0 to N as X and display J[X]'s getId()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Job

{

    int id;

    char name;

    int time, space;

public:

    void readData()

    {

        cin >> id >> name >> time >> space;

    }

    int getTime()

    {

        return time;

    }

    int getSpace()

    {

        return space;

    }

    int getId()

    {

        return id;

    }

};

class OS

{

public:

    void sortTime(int n, Job \*j)

    {

        Job temp;

        for (int x = 0; x < n; x++)

            for (int y = x + 1; y < n; y++)

                if (j[x].getTime() > j[y].getTime())

                {

                    temp = j[x];

                    j[x] = j[y];

                    j[y] = temp;

                }

    }

    void sortSpace(int n, Job \*j)

    {

        Job temp;

        for (int x = 0; x < n; x++)

            for (int y = x + 1; y < n; y++)

                if (j[x].getSpace() > j[y].getSpace())

                {

                    temp = j[x];

                    j[x] = j[y];

                    j[y] = temp;

                }

    }

};

int main()

{

    int n;

    char sort\_type[10];

    cin >> n >> sort\_type;

    Job j[n];

    for (int x = 0; x < n; x++)

        j[x].readData();

    OS OS1\_X;

    if (strcmp(sort\_type, "Time") == 0)

        OS1\_X.sortTime(n, j);

    else if (strcmp(sort\_type, "Space") == 0)

        OS1\_X.sortSpace(n, j);

    for (int x = 0; x < n; x++)

        cout << j[x].getId() << "\n";

}

### Output

Graphical user interface, treemap chart

Description automatically generated with medium confidence

### [Ex. No. M7\_CSQ7]

### AIM

Create an abstract class called Shape with two data members height and weight. Define a member function named SetValue and a pure virtual function name Area. Inherit this abstract class and create two children named Rectangle and Triangle. Implement Area in child classes.

Note: Area of Rectangle = h \* w, Area of Triangle = (h\*w)/2

### Algorithm / Pseudocode

Shape:

Declare integers Height and Width

Void setValues(int H, int W):

Assign Height as H

Assign Width as W

Declare pure virtual int area()

Rectangle, extends Shape:

Int area():

Return Height \* Width

Triangle, extends Shape:

Int area():

Return Height \* Width / 2

Int main():

Declare integer Choice, Height and Width

Read inputs and assign to Choice, Height and Width

Declare Shape pointer S

Switch Choice,

If 1, then declare Rectangle R and assign its address to S

If 2, then declare Triangle T and assign its address to S

Call S's setValues() and pass Height and Width

Display S'x area()

### Program Code

// 21BDS0340 Abhinav Dinesh Srivatsa

#include <iostream>

#include <string.h>

using namespace std;

class Shape

{

protected:

    int height, width;

public:

    void setValues(int height, int width)

    {

        this->height = height;

        this->width = width;

    }

    virtual int area() = 0;

};

class Rectangle : public Shape

{

public:

    int area()

    {

        return height \* width;

    }

};

class Triangle : public Shape

{

public:

    int area()

    {

        return height \* width / 2;

    }

};

int main()

{

    int choice;

    cin >> choice;

    int height, width;

    cin >> height >> width;

    Shape \*s;

    switch (choice)

    {

    case 1:

    {

        Rectangle r;

        s = &r;

        break;

    }

    case 2:

    {

        Triangle t;

        s = &t;

        break;

    }

    }

    s->setValues(height, width);

    cout << s->area();

}

### Output

Graphical user interface

Description automatically generated

### [Ex. No. M7\_CSQ8]

### AIM