|  |
| --- |
|  |
| EXPERIMENT 3 |
|  |
|  |
| **Vansh Sukhija**  **12112021** |
|  |

|  |
| --- |
|  |

Ans 1

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void main(){

    char \*ptr = (char \*)malloc(15\*sizeof(char));

    strcpy(ptr, "Hello World");

    puts(ptr);

    ptr = (char \*)realloc(ptr, 30);

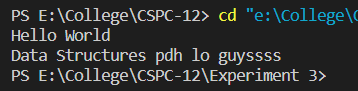
    strcpy(ptr, "Data Structures pdh lo guyssss");

    puts(ptr);

    free(ptr);

}

Output:



Ans 2:

#include<stdio.h>

#include<stdlib.h>

int main(){

    int r1, c1, r2, c2;

    printf("Enter the index of 1st matrix: ");

    scanf("%d %d", &r1, &c1);

    printf("Enter the index of 2nd matrix: ");

    scanf("%d %d", &r2, &c2);

    if(c1!=r2){

        printf("Cant be multiplied");

        return 0;

    }

    else{

        int \*mat1 = (int \*)malloc(r1\*c1\*sizeof(int));

        int \*mat2 = (int \*)malloc(r2\*c2\*sizeof(int));

        int \*mat3 = (int \*)malloc(r1\*c2\*sizeof(int));

        printf("Enter 1st matrix:\n");

        for(int i=0; i<r1\*c1; i++){

            scanf("%d", mat1+i);

        }

        printf("Enter 2nd matrix:\n");

        for(int i=0; i<r2\*c2; i++){

            scanf("%d", mat2+i);

        }

        for(int i=0; i<r1; i++){

            for(int j=0; j<c2; j++){

                \*(mat3+i\*r1+j) = 0;

                for(int k=0; k<c1; k++){

                    \*(mat3+i\*r1+j) += (\*(mat1+i\*r1+k))\*(\*(mat2+k\*c2+j));

                }

            }

        }

        printf("Multiplied Matrix:\n");

        for(int i=0; i<r1; i++){

            for(int j=0; j<c2; j++){

                printf("%d ", \*(mat3+i\*r1+j));

            }

            printf("\n");

        }

        free(mat1);

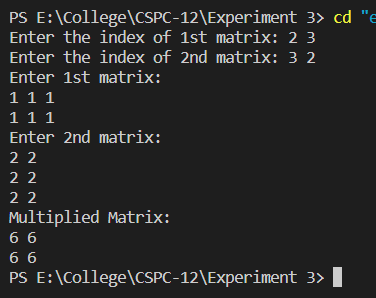
        free(mat2);

        free(mat3);

    }

}

Output:



Ans 3:

#include<stdio.h>

#include<stdlib.h>

typedef struct student{

    char name[30];

    int roll\_no;

    int marks;

}stud;

static int count=1;

void first(stud \*ptr, int n){

    if(count>n)

        printf("Cannot enter more data\n");

    else{

        fflush(stdin);

        printf("Enter name of student: ");

        scanf("%[^\n]s", &(ptr+count-1)->name);

        printf("Enter Roll no.: ");

        scanf("%d", &(ptr+count-1)->roll\_no);

        printf("Enter total marks: ");

        scanf("%d", &(ptr+count-1)->marks);

        count++;

    }

    printf("\n");

}

void second(stud \*ptr, int n){

    for(int i=0; i<n-1; i++){

        printf("\n");

        puts((ptr+i)->name);

        printf("%d\n", (ptr+i)->roll\_no);

        printf("%d\n", (ptr+i)->marks);

    }

    printf("\n");

}

int main(){

    int n;

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

    scanf("%d", &n);

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

    while(1){

        int input, num;

        printf("Enter 0 to exit\nEnter 1 to input data\nEnter 2 to display data\n");

        scanf("%d", &input);

        switch(input){

            case 0:

                return 0;

            case 1:

                first(ptr, n);

                break;

            case 2:

                second(ptr, count);

                break;

            default:

                printf("Wrong input\n");

                break;

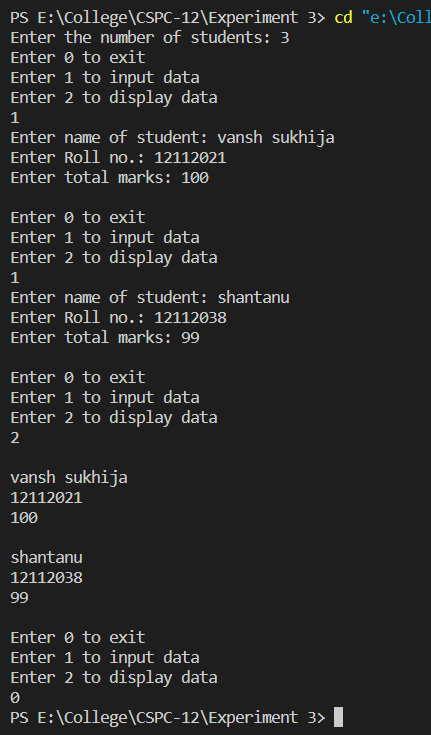
        }

    }

    free(ptr);

}

Output:



Ans 4:

#include<stdio.h>

#include<stdlib.h>

typedef struct employee{

    char name[30];

    int faculty\_id;

    int salary;

}emp;

static int count=1;

void one(emp \*ptr, int n){

    if(count>n)

        printf("Cannot enter more data\n");

    else{

        fflush(stdin);

        printf("Enter name of employee: ");

        scanf("%[^\n]s", &(ptr+count-1)->name);

        printf("Enter employee no.: ");

        scanf("%d", &(ptr+count-1)->faculty\_id);

        printf("Enter salary: ");

        scanf("%d", &(ptr+count-1)->salary);

        count++;

    }

    printf("\n");

}

void two(emp \*ptr, int n){

    for(int i=0; i<n-1; i++){

        printf("\n");

        puts((ptr+i)->name);

        printf("%d\n", (ptr+i)->faculty\_id);

        printf("%d\n", (ptr+i)->salary);

    }

    printf("\n");

}

int main(){

    int n;

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

    scanf("%d", &n);

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

    while(1){

        int input, num;

        printf("Enter 0 to exit\nEnter 1 to input data\nEnter 2 to display data\n");

        scanf("%d", &input);

        switch(input){

            case 0:

                return 0;

            case 1:

                one(ptr, n);

                break;

            case 2:

                two(ptr, count);

                break;

            default:

                printf("Wrong input\n");

                break;

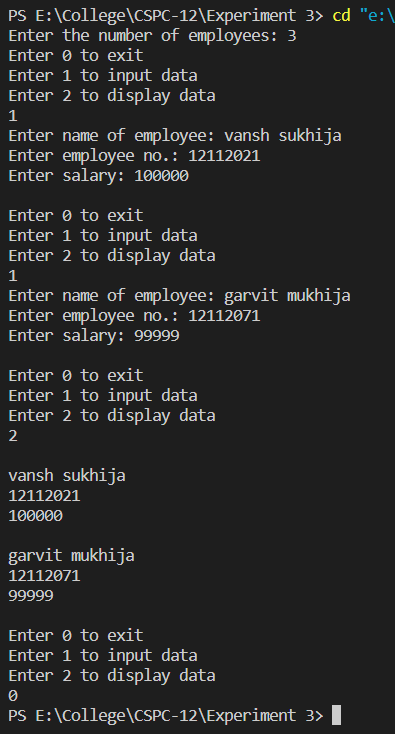
        }

    }

    free(ptr);

}

Output:



Ans 5:

#include<stdio.h>

#include<stdlib.h>

typedef struct faculty{

    char name[30];

    int faculty\_id;

    char code[7];

    char class\_names[4];

}fac;

static int count=1;

void func1(fac \*ptr, int n){

    if(count>n)

        printf("Cannot enter more data\n");

    else{

        fflush(stdin);

        printf("Enter name of faculty: ");

        scanf("%[^\n]s", &(ptr+count-1)->name);

        printf("Enter faculty ID: ");

        scanf("%d", &(ptr+count-1)->faculty\_id);

        fflush(stdin);

        printf("Enter course code: ");

        gets((ptr+count-1)->code);

        fflush(stdin);

        printf("Enter Class names: ");

        gets((ptr+count-1)->class\_names);

        count++;

    }

    printf("\n");

}

void func2(fac \*ptr, int n){

    for(int i=0; i<n-1; i++){

        printf("\n");

        puts((ptr+i)->name);

        printf("%d\n", (ptr+i)->faculty\_id);

        puts((ptr+i)->code);

        puts((ptr+i)->class\_names);

    }

    printf("\n");

}

int main(){

    int n;

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

    scanf("%d", &n);

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

    while(1){

        int input, num;

        printf("Enter 0 to exit\nEnter 1 to input data\nEnter 2 to display data\n");

        scanf("%d", &input);

        switch(input){

            case 0:

                return 0;

            case 1:

                func1(ptr, n);

                break;

            case 2:

                func2(ptr, count);

                break;

            default:

                printf("Wrong input\n");

                break;

        }

    }

    free(ptr);

}

Output:

