//Q1

/\*Allocate memory for 10 integers, and input 10 integers from user. Print the second maximum and second minimum values and de-allocate the memory you created above.\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int \*ptr;

    ptr = (int \*)malloc(10 \* sizeof(int));

    if (ptr == NULL)

    {

        printf("Memory not allocated.\n");

        exit(0);

    }

    else

    {

        printf("Memory successfully allocated using malloc.\n");

        for (int i = 0; i < 10; i++)

        {

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

            scanf("%d", &ptr[i]);

        }

        int max = ptr[0], min = ptr[0], second\_max = ptr[0], second\_min = ptr[0];

        for (int i = 0; i < 10; i++)

        {

            if (ptr[i] > max)

            {

                second\_max = max;

                max = ptr[i];

            }

            else if (ptr[i] > second\_max && ptr[i] != max)

            {

                second\_max = ptr[i];

            }

            if (ptr[i] < min)

            {

                second\_min = min;

                min = ptr[i];

            }

            else if (ptr[i] < second\_min && ptr[i] != min)

            {

                second\_min = ptr[i];

            }

        }

        printf("Second maximum value is: %d\n", second\_max);

        printf("Second minimum value is: %d\n", second\_min);

        free(ptr);

    }

    return 0;

}

//Q2

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*arr;

int n, sum = 0;

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

scanf("%d", &n);

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

if (arr == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter %d elements:\n", n);

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

scanf("%d", &arr[i]);

sum += arr[i];

}

printf("Sum of elements: %d\n", sum);

free(arr);

return 0;

}

//Q3

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*arr;

int n, largest;

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

scanf("%d", &n);

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

if (arr == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter %d elements:\n", n);

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

scanf("%d", &arr[i]);

}

largest = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] > largest) {

largest = arr[i];

}

}

printf("The largest element in the array is: %d\n", largest);

free(arr);

return 0;

}

//Q4

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

char \*str, \*rev;

int len;

printf("Enter a string: ");

scanf("%ms", &str); // %ms is used to dynamically allocate memory for input string

len = strlen(str);

rev = (char \*)malloc((len + 1) \* sizeof(char));

if (rev == NULL) {

printf("Memory allocation failed.\n");

free(str);

return 1;

}

for (int i = 0; i < len; i++) {

rev[i] = str[len - i - 1];

}

rev[len] = '\0'; // Null-terminate the reversed string

printf("Reversed string: %s\n", rev);

free(str);

free(rev);

return 0;

}

//Q5

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*arr1, \*arr2;

int size;

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

scanf("%d", &size);

arr1 = (int \*)malloc(size \* sizeof(int));

if (arr1 == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter %d elements of the array:\n", size);

for (int i = 0; i < size; i++) {

scanf("%d", &arr1[i]);

}

arr2 = (int \*)malloc(size \* sizeof(int));

if (arr2 == NULL) {

printf("Memory allocation failed.\n");

free(arr1);

return 1;

}

// Copy elements from arr1 to arr2

for (int i = 0; i < size; i++) {

arr2[i] = arr1[i];

}

printf("Copied array:\n");

for (int i = 0; i < size; i++) {

printf("%d ", arr2[i]);

}

printf("\n");

free(arr1);

free(arr2);

return 0;

}

//Q6

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*arr;

int size, newSize;

printf("Enter the initial size of the array: ");

scanf("%d", &size);

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

if (arr == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter %d elements of the array:\n", size);

for (int i = 0; i < size; i++) {

scanf("%d", &arr[i]);

}

printf("Enter the new size of the array: ");

scanf("%d", &newSize);

arr = (int \*)realloc(arr, newSize \* sizeof(int));

if (arr == NULL) {

printf("Memory reallocation failed.\n");

return 1;

}

printf("Enter %d more elements of the array:\n", newSize - size);

for (int i = size; i < newSize; i++) {

scanf("%d", &arr[i]);

}

printf("Array elements after resizing:\n");

for (int i = 0; i < newSize; i++) {

printf("%d ", arr[i]);

}

printf("\n");

free(arr);

return 0;

}

//Q7

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main() {

char \*str;

char ch;

int count = 0;

// Allocate memory for the string dynamically

str = (char \*)malloc(100 \* sizeof(char)); // Assuming maximum length of 100 characters

if (str == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter a string: ");

scanf("%s", str);

printf("Enter a character to count: ");

scanf(" %c", &ch); // Note the space before %c to consume any leading whitespace

// Count occurrences of the character in the string

for (int i = 0; i < strlen(str); i++) {

if (str[i] == ch) {

count++;

}

}

printf("Occurrences of '%c' in the string: %d\n", ch, count);

// Free dynamically allocated memory

free(str);

return 0;

}

//Q8

#include <stdio.h>

#include <stdlib.h>

int main() {

int \*numbers, n;

float sum = 0, average;

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

scanf("%d", &n);

// Allocate memory for array dynamically

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

if (numbers == NULL) {

printf("Memory allocation failed.\n");

return 1;

}

printf("Enter %d numbers:\n", n);

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

scanf("%d", &numbers[i]);

sum += numbers[i];

}

// Calculate average

average = sum / n;

printf("Average of %d numbers: %.2f\n", n, average);

// Free dynamically allocated memory

free(numbers);

return 0;

}

//Q1 struct

//struct1

#include<stdio.h>

#include<string.h>

typedef struct student

{

    char nm[30];

    int age;

    int m;

}stu;

int main()

{

    int n,i,j,sum = 0,avg;

    stu arr[30];

    printf("Enter the number of students to store the record of: \n");

    scanf("%d",&n);

    printf("Enter the details of the students:\n");

    for(i = 0; i < n; i++)

    {

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

        //fgets(arr[i].nm,sizeof(arr[i].nm),stdin);

        scanf("%s",arr[i].nm);

        printf("Enter the age of student %d : ",i+1);

        scanf("%d",&arr[i].age);

        printf("Enter the Marks of student %d in single digit : ",i+1);

        scanf("%d",&arr[i].m);

    }

    printf("The details of student are as follows :\n");

    printf ("Name   Age Total Marks\n");

    for(i = 0;i < n; i++)

    {

        fputs(arr[i].nm,stdout);

        printf("\t");

        printf("%d  %d",arr[i].age,arr[i].m);

        printf("\n");

    }

    for(i = 0; i < n; i++)

    {

        sum += arr[i].m;

    }

    avg = sum/n;

    printf("The average marks of all the %d students are : %d\n",n,avg);

    return 0;

}

//Q2 struct

//struct2

#include<stdio.h>

#include<string.h>

typedef struct book

{

    char t[30];

    char auth[30];

    int p;

}book;

int main()

{

    int n,i,j,max,min;

    book arr[30];

    printf("Enter the number of books to store the record of: \n");

    scanf("%d",&n);

    printf("Enter the details of the books:\n");

    for(i = 0; i < n; i++)

    {

        printf("Enter the title of the book %d : ",i+1);

        //fgets(arr[i].nm,sizeof(arr[i].nm),stdin);

        scanf("%s",arr[i].t);

        printf("Enter the Author of the book %d : ",i+1);

        scanf("%s",arr[i].auth);

        printf("Enter the price of book %d : ",i+1);

        scanf("%d",&arr[i].p);

    }

    printf("The details of books are as follows :\n");

    printf ("Title  Author  Price\n");

    for(i = 0;i < n; i++)

    {

        fputs(arr[i].t,stdout);

        printf("\t");

        fputs(arr[i].auth,stdout);

        printf("\t");

        printf("%d",arr[i].p);

        printf("\n");

    }

    max = 0;

    min = 0;

    for(i = 0; i < n; i++)

    {

        if(arr[i].p>arr[max].p)

        {

            max = i;

        }

        if(arr[i].p<arr[min].p)

        {

            min = i;

        }

    }

    printf("The Most expensive book is :\n");

    printf ("Title  Author  Price\n");

    fputs(arr[max].t,stdout);

    printf("\t");

    fputs(arr[max].auth,stdout);

    printf("\t");

    printf("%d",arr[max].p);

    printf("\n");

    printf("The Cheapest book is :\n");

    printf ("Title  Author  Price\n");

    fputs(arr[min].t,stdout);

    printf("\t");

    fputs(arr[min].auth,stdout);

    printf("\t");

    printf("%d",arr[min].p);

    printf("\n");

    return 0;

}

//Q3 struct

//struct3

#include<stdio.h>

typedef struct circle

{

    float rad;

    float ar;

    float peri;

}circ;

int main()

{

    int n,i,j;

    circ arr[30];

    printf("Enter the number of Circles to store the record of: \n");

    scanf("%d",&n);

    printf("Enter the radius of circles:\n");

    for(i = 0; i < n; i++)

    {

        printf("Enter the radius of circle %d : ",i+1);

        scanf("%f",&arr[i].rad);

    }

    //Calculating the area and perimeter of the circle

    for(i = 0; i < n; i++)

    {

        arr[i].ar = 3.14\*(arr[i].rad \* arr[i].rad);

        arr[i].peri = 2 \* 3.14 \* arr[i].rad;

    }

    printf("The details of Circles are as follows :\n");

    printf ("Circle Radius  Area    Circumference\n");

    for(i = 0;i < n; i++)

    {

        printf("%d  %.4f    %.4f    %.4f",i+1,arr[i].rad,arr[i].ar,arr[i].peri);

        printf("\n");

    }

    return 0;

}

//Q4 struct

#include<stdio.h>

#include<string.h>

typedef struct car

{

    int id;

    int model;

    int rrprice;

} car;

int main()

{

    int n,i,j,sum = 0,day,total;

    car arr[30];

    printf("Enter the number of cars to store the record of: \n");

    scanf("%d",&n);

    printf("Enter the details of the cars:\n");

    for(i = 0; i < n; i++)

    {

        printf("Enter the id of car %d : ",i+1);

        scanf("%d",&arr[i].id);

        printf("Enter the model of car %d : ",i+1);

        scanf("%d",&arr[i].model);

        printf("Enter the Road Runner Price of car %d : ",i+1);

        scanf("%d",&arr[i].rrprice);

    }

    printf("The details of cars are as follows :\n");

    printf ("ID Model   Road Runner Price\n");

    for(i = 0;i < n; i++)

    {

        printf("%d  %d  %d",arr[i].id,arr[i].model,arr[i].rrprice);

        printf("\n");

    }

    printf("Enter the number of days to calculate the average Road Runner Price of all the cars: \n");

    scanf("%d",&day);

    for(i = 0; i < n; i++)

    {

        sum += arr[i].rrprice;

    }

    total = sum \* day;

    printf("The Total Rental Rate of all the %d cars are : %d\n",n,total);

    return 0;

}