Assignment 20

DMA

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void inputString(char \*\*str) {

\*str = (char \*)malloc(sizeof(char));

printf("Enter a string: ");

scanf(" %s", \*str);

}

int main() {

//1. Define a function to input variable-length string and store it in an array without memory wastage:

char \*input;

inputString(&input);

printf("You entered: %s\n\n", input);

free(input);

//2. Program to calculate the average of data values:

int n, \*arr, sum = 0;

printf("Enter the number of data values: ");

scanf("%d", &n);

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

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

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

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

sum += arr[i];

}

printf("Average of data values: %.2f\n\n", (float)sum / n);

free(arr);

//3. Program to calculate the sum of n numbers using malloc and free:

sum = 0, n = 0;

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

scanf("%d", &n);

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

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

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

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

sum += arr[i];

}

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

free(arr);

//4. Program to input and print text using dynamic memory allocation:

char \*text;

printf("Enter text: ");

text = (char \*)malloc(100 \* sizeof(char)); // assuming max length of text is 99 characters

scanf(" %s", text);

printf("You entered: %s\n\n", text);

free(text);

//5. Program to read a one-dimensional array, print sum of all elements along with inputted array elements:

sum = 0, n = 0;

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

scanf("%d", &n);

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

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

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

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

sum += arr[i];

}

printf("Array elements: ");

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

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

}

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

free(arr);

//6. Program to find the largest element using Dynamic Memory Allocation:

int max;

sum = 0, n = 0;

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

scanf("%d", &n);

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

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

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

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

}

max = arr[0];

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

if (arr[i] > max) {

max = arr[i];

}

}

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

free(arr);

//7. Program to demonstrate memory leak in C:

// Memory leak example

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

//8. Program to demonstrate dangling pointers in C:

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

\*ptr = 5;

free(ptr);

// Dereferencing a dangling pointer

printf("%d\n\n", \*ptr);

//9. Program to allocate memory dynamically with error handling:

int size;

printf("Enter the size in bytes: ");

scanf("%d", &size);

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

if (arr == NULL) {

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

} else {

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

free(arr);

}

//10. Program to find the maximum and minimum from an array using dynamic memory allocation:

int min;

n = 0, max = 0;

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

scanf("%d", &n);

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

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

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

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

}

max = min = arr[0];

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

if (arr[i] > max) {

max = arr[i];}

if (arr[i] < min) {

min = arr[i];}

}

printf("Maximum element: %d\n", max);

printf("Minimum element: %d\n", min);

free(arr);

return 0;

}



