Assignment - 22 A Job Ready Bootcamp in C++, DSA and IOT MySirG

DMA

1. Define a function to input variable length string and store it in an array without

memory wastage.

#include <stdio.h>

#include <stdlib.h>

char\* inputVarLengthString()

{

int varLength;

printf("Enter variable length: ");

scanf("%d", &varLength);

char \*ptr = (char \*)calloc(varLength, sizeof(char));

printf("Input %d length string: ", varLength);

fflush(stdin);

fgets(ptr, varLength \* sizeof(char) + 1, stdin); // +1 because 1 character hold null character

return ptr;

}

int main()

{

char \*ptr;

ptr = inputVarLengthString();

printf("You enter string - %s", ptr);

return 0;

}

2. Write a program to ask the user to input a number of data values he would like to

enter then create an array dynamically to accommodate the data values. Now take

the input from the user and display the average of data values.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int countOfData, sumOfValues = 0;

float averageOfValues;

printf("input a number of data values you would like to enter: ");

scanf("%d", &countOfData);

int \*ptr;

// ptr = (int \*)calloc(countOfData, sizeof(int));

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

for (int i = 0; i < countOfData; i++)

{

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

scanf("%d", (ptr + i));

sumOfValues += \*(ptr + i);

}

averageOfValues = sumOfValues / (float)countOfData;

printf("Average of data values: %.2f", averageOfValues);

free(ptr);

return 0;

}

3. Write a program to calculate the sum of n numbers entered by the user using malloc

and free.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n, sum = 0;

printf("How many element u want to enter: ");

scanf("%d", &n);

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

printf("Input %d Element: ", n);

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

{

printf("\nEnter %d Element: ", i + 1);

scanf("%d", (ptr + i));

sum += \*(ptr + i);

}

printf("\nsum is %d", sum);

return 0;

}

4. Write a program to input and print text using dynamic memory allocation.

#include <stdio.h>

#include <stdlib.h>

char \*inputTEXT()

{

int txtLength;

printf("input length of text: ");

scanf("%d", &txtLength);

fflush(stdin);

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

printf("Input Text: ");

fgets(ptr, txtLength \* sizeof(char), stdin);

return ptr;

}

void outputText(char \*ptr)

{

printf("Output Text: %s", ptr);

}

int main()

{

char \*ptr;

ptr = inputTEXT();

outputText(ptr);

return 0;

}

5. Write a program to read a one dimensional array, print sum of all elements along with

inputted array elements using dynamic memory allocation.

#include <stdio.h>

#include <stdlib.h>

int sumOfNValues(int n)

{

int sum = 0;

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

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

{

printf("Enter %d Values: ", i + 1);

scanf("%d", (ptr + i));

sum += \*(ptr + i);

}

free(ptr); // relise dma memory

return sum;

}

int main()

{

int n;

printf("How many values you want to enter: ");

scanf("%d", &n);

int sum = sumOfNValues(n);

printf("\nSum of %d Values is: %d", n, sum);

return 0;

}

6. Write a program in C to find the largest element using Dynamic Memory Allocation.

#include <stdio.h>

#include <stdlib.h>

int \*inputElement(int n)

{

printf("Input %d Element: ", n);

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

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

{

printf("Enter %d Element: ", i + 1);

scanf("%d", (ptr + i));

}

return ptr;

}

int largestElement(int \*ptr, int n)

{

int largestElement = -1;

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

{

if (largestElement < \*(ptr + i))

{

largestElement = \*(ptr + i);

}

}

return largestElement;

}

int main()

{

int n;

printf("How many element u want to enter: ");

scanf("%d", &n);

int \*ptr;

ptr = inputElement(n);

int large = largestElement(ptr, n);

printf("%d is largest Element", large);

return 0;

}

7. Write a program to demonstrate memory leak in C.

#include <stdio.h>

#include <stdlib.h>

void memoryLeak()

{

int \*ptr = (int \*)malloc(sizeof(int)); // ptr contain address of 4 bytes memory

}

int main()

{

memoryLeak();

return 0;

}

8. Write a program to demonstrate dangling pointers in C.

#include<stdio.h>

int main(){

int \*ptr;

{

int x = 5;

ptr = &x;

}

// now ptr become dangling pointer becusae ptr contains address of x but x is destroy but ptr contain x address

return 0;

}

9. Write a program to allocate memory dynamically of the size in bytes entered by the

user. Also handle the case when memory allocation is failed.

#include <stdio.h>

#include <stdlib.h>

int main()

{

int bytes;

printf("Enter size in bytes for allocating memory dymanically: ");

scanf("%d", &bytes);

int \*ptr = NULL;

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

if (ptr != NULL)

printf("Memory Allocated Successfull!");

else

printf("Memory Error!");

return 0;

}

10. Find out the maximum and minimum from an array using dynamic memory allocation

in C.

#include <stdio.h>

#include <stdlib.h>

int \*inputElement(int n)

{

printf("Input %d Element: ", n);

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

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

{

printf("Enter %d Element: ", i + 1);

scanf("%d", (ptr + i));

}

return ptr;

}

int minAndMaxElement(int \*ptr, int n)

{

int max = -1;

int min = \_\_INT\_MAX\_\_;

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

{

if (max < \*(ptr + i))

max = \*(ptr + i);

if (min > \*(ptr + i))

min = \*(ptr + i);

}

printf("Min Element - %d, Max Element - %d", min, max);

}

int main()

{

int n;

printf("How many element u want to enter: ");

scanf("%d", &n);

int \*ptr;

ptr = inputElement(n);

minAndMaxElement(ptr, n);

return 0;

}