**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>

int main()

{

char \*str, c;

int i=0, j=1;

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

printf("Enter string: ");

while(c!='\n')

{

c=getc(stdin);

j++;

str=(char\*)realloc(str, j\*sizeof(char));

str[i]=c;

i++;

}

str[i]='\0';

printf("\n The entered string is: %s",str);

free(str);

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>

int main()

{

int \*ptr, c, i=0, size=0, sum=0;

printf("Enter size of array: ");

scanf("%d",&size);

ptr=(int\*)calloc(size, sizeof(int));

if(ptr==NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("\n The entered %d values\n",size);

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

{

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

}

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

{

sum=sum+\*(ptr+i);

}

printf("Average is %d\n",sum/size);

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>

int main()

{

int \*ptr, c, i=0, size=0, sum=0;

printf("Enter size of array: ");

scanf("%d",&size);

ptr=(int\*)calloc(size, sizeof(int));

if(ptr==NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("\n The entered %d values\n",size);

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

{

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

}

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

{

sum=sum+\*(ptr+i);

}

printf("Average is %d\n",sum/size);

free(ptr);

return 0;

}

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

#include <stdio.h>

#include <stdlib.h>

int main() {

int size;

char \*text;

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

scanf("%d", &size);

text = (char\*)malloc(size \* sizeof(char));

if (text == NULL) {

printf("Memory allocation failed!");

return 0;

}

printf("Enter the text: ");

scanf(" %[^\n]s", text); // read text including spaces

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

free(text); // free dynamically allocated memory

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 main() {

int size, i, sum = 0;

int \*arr;

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

scanf("%d", &size);

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

if (arr == NULL) {

printf("Memory allocation failed!");

return 0;

}

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

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

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

sum += \*(arr + i);

}

printf("The entered array elements are:\n");

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

printf("%d ", \*(arr + i));

}

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

free(arr);

return 0;

}

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

#include <stdio.h>

#include <stdlib.h>

int main() {

int size, i;

int \*arr;

int max;

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

scanf("%d", &size);

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

if (arr == NULL) {

printf("Memory allocation failed!");

return 0;

}

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

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

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

}

max = \*arr;

for (i = 1; i < size; i++) {

if (\*(arr + i) > max) {

max = \*(arr + i);

}

}

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

free(arr);

return 0;

}

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

#include <stdio.h>

#include <stdlib.h>

int main()

{

int \*ptr;

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

ptr=NULL;

return 0;

}

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

#include <stdio.h>

int main()

{

int \*ptr;

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

\*ptr=10;

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

free(ptr);

printf("After free %d",\*ptr);

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 size;

void \*ptr;

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

scanf("%d", &size);

ptr = malloc(size);

if (ptr == NULL) {

printf("Memory allocation failed!");

return 0;

}

free(ptr);

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 main() {

int \*arr, size, i, max, min;

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

scanf("%d", &size);

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

if (arr == NULL) {

printf("Memory allocation failed!");

return 0;

}

printf("Enter the elements of the array:\n");

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

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

}

max = min = \*arr;

for (i = 1; i < size; i++) {

if (\*(arr + i) > max) {

max = \*(arr + i);

}

if (\*(arr + i) < min) {

min = \*(arr + i);

}

}

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

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

free(arr);

return 0;

}