**Assignment - 22**

**A Job Ready Bootcamp in C++, DSA and IOT**

**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("\nThe entered string is : %s",str);

printf("\nLength is %d",--i);

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;

int size=0,i,sum=0;

printf("Enter size of array\n");

scanf("%d",&size);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Enter %d number : ",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",sum/size);

free(ptr);

}

1. 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;

int n,i,sum=0;

printf("How many number you want to add\n");

scanf("%d",&n);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Enter %d number : ",n);

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

{

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

}

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

{

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

}

printf("Sum is %d",sum);

free(ptr);

}

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

#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("\nThe entered string is : %s",str);

free(str);

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>

int main()

{

int \*ptr;

int n,i,sum=0;

printf("How many number you want to add\n");

scanf("%d",&n);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Enter %d number : ",n);

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

{

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

}

printf("Entered element is\n");

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

{

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

}

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

{

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

}

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

free(ptr);

}

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

#include<stdio.h>

int main()

{

int \*ptr;

int n,i,max=-1;

printf("How many number you want to add\n");

scanf("%d",&n);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Enter %d number : ",n);

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

{

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

}

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

{

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

{

max=\*(ptr+i);

}

}

printf("Largest element in the array is %d",max);

free(ptr);

}

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

#include<stdio.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”,\*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>

int main()

{

int \*ptr;

int n;

printf("Enter size of array\n");

scanf("%d",&n);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Memory allocation is not fail");

return 0;

}

OutputEnter size of array 100000000000000

Memory Allocation Failed

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

#include<stdio.h>

int main()

{

int \*ptr;

int n,i,max=-1,min=99999999;

printf("How many number you want to add\n");

scanf("%d",&n);

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

if(ptr == NULL)

{

printf("Memory Allocation Failed\n");

return 0;

}

printf("Enter %d number : ",n);

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

{

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

}

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

{

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

{

max=\*(ptr+i);

}

}

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

{

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

{

min=\*(ptr+i);

}

}

printf("Largest element in the array is %d",max);

printf("\nSmallest element in the array is %d",min);

free(ptr);

return 0;

}