**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.

Sol – 1.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i=0,j=0;

char c=0,\*ptr=NULL;

ptr=(char\*)malloc(0);

printf("Enter string : ");

while(c!='\n')

{

c=getc(stdin);

j++;

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

ptr[i]=c;

i++;

}

ptr[i-1]=0;

printf("Entered string : %s",ptr);

getch();

return 0;

}

OR

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

int main()

{

int i,n,m;

printf("Enter how much strings : ");

scanf("%d",&n);

char\* ptr[n];

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

{

printf("Enter size of string : ");

fflush(stdin);

scanf("%d",&m);

ptr[i]=(char\*)malloc(m+1);

printf("Enter the string : ");

fflush(stdin);

fgets(ptr[i],m+1,stdin);

// ptr[i][strlen(ptr[i])]=0;

}

printf("\n\nVariable Length Strings are\n\n");

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

{

printf("%s\n",ptr[i]);

}

free(ptr);

getch();

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.

Sol – 2.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i,n,\*p,sum=0;

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

scanf("%d",&n);

p=(int\*)calloc(n,sizeof(int));

if(p==NULL)

{

printf("Memory Allocation Failed");

return 0;

}

printf("Enter data\n");

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

{

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

}

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

{

sum=sum+p[i];

}

printf("\nAverage : %f",sum\*1.0/n);

free(p);

getch();

return 0;

}

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

and free.

Sol – 3.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i,n,\*p,sum=0;

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

scanf("%d",&n);

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

if(p==NULL)

{

printf("Memory Allocation Failed");

return 0;

}

printf("Enter data\n");

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

{

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

}

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

{

sum=sum+p[i];

}

free(p);

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

getch();

return 0;

}

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

Sol – 4.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i=0,j=0;

char c=0,\*ptr=NULL;

ptr=(char\*)malloc(0);

printf("Enter Text : ");

while(c!='\n')

{

c=getc(stdin);

j++;

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

ptr[i]=c;

i++;

}

ptr[i-1]=0;

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

getch();

return 0;

}

OR

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

int main()

{

char\* p;

printf("Enter text : ");

p=(char\*)calloc(100,sizeof(char));

fgets(p,100,stdin);

p[strlen(p)-1]=0;

printf("Your text : %s",p);

getch();

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.

Sol – 5.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i,n,\*p,sum=0;

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

scanf("%d",&n);

p=(int\*)malloc(n\*4);

if(p==NULL)

{

printf("Memory allocation failed");

return 0;

}

printf("Enter data\n");

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

{

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

}

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

{

sum=sum+p[i];

}

printf("\nAll elements\n");

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

{

printf("%d\n",p[i]);

}

free(p);

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

getch();

return 0;

}

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

Sol – 6.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i,j,n,\*p;

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

scanf("%d",&n);

p=(int\*)calloc(n,sizeof(int));

if(p==NULL)

{

printf("Memory allocation failed");

return 0;

}

printf("Enter data\n");

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

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

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

{

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

{

if(p[i]<p[j])

break;

}

if(j==n)

{

printf("\n%d is largest",p[i]);

free(p);

break;

}

}

getch();

return 0;

}

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

Sol – 7.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

void f();

int main()

{

f();

printf("\nHo gyi memory leak as we don't have address of variable we created using malloc function");

getch();

return 0;

}

void f()

{

char\* p;

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

\*p='A';

printf("%c",\*p);

}

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

Sol – 8.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

char\* p;

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

\*p='A';

printf("%c",\*p);

free(p);

printf("\nHere P is dangling pointer as it is pointing an address which is not reserved by us");

getch();

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.

Sol – 9.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int n;

void\* p;

printf("Enter how much bytes you want to reserve : ");

scanf("%d",&n);

p=malloc(n);

if(p==NULL)

{

printf("Memory allocation failed");

return 0;

}

printf("Memory reserved successfully");

getch();

return 0;

}

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

in C.

Sol – 10.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

int main()

{

int i,j,n,\*p,l,s;

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

scanf("%d",&n);

p=(int\*)calloc(n,sizeof(int));

if(p==NULL)

{

printf("Memory allocation failed");

return 0;

}

printf("Enter data\n");

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

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

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

{

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

{

if(p[i]<p[j])

break;

}

if(j==n)

{

l=p[i];

break;

}

}

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

{

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

{

if(p[i]>p[j])

break;

}

if(j==n)

{

s=p[i];

break;

}

}

free(p);

printf("Smallest : %d and Largest : %d",s,l);

getch();

return 0;

}