**Assignment-22 Solution Name: OM PANT**

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

Ans-

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

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main(){

    int length,i;

    char \*p;

    printf("Enter length of string\n");

    scanf("%d",&length);

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

    printf("Enter the string\n");

    for(i=0;i<length;i++){

        scanf("%c",p+i);

    }

    printf("String Entered...\n");

    for(i=0;i<length;i++){

        printf("%c", \*(p+i));

    }

    free(p);

    return 0;

}

1. 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.

Ans-

// 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>

#include<string.h>

int main(){

    int length,\*p,i;

    printf("Enter no. of data values\n");

    scanf("%d",&length);

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

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

    for(i=0;i<length;i++){

        scanf("%d",p+i);

    }

    int sum=0,average = 0;

    for(i=0;i<length;i++){

        sum += \*(p+i);

    }

    average = sum/length;

    printf("Average of Values: %d",average);

    free(p);

    return 0;

}

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

Ans-

// 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>

#include<string.h>

int main(){

    int length,\*p,i;

    printf("Enter no. of values\n");

    scanf("%d",&length);

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

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

    for(i=0;i<length;i++){

        scanf("%d",p+i);

    }

    int sum=0;

    for(i=0;i<length;i++){

        sum += \*(p+i);

    }

    printf("Sum of Values: %d",sum);

    free(p);

    return 0;

}

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

Ans-

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

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main(){

    int size;

    printf("Enter no of characters\n");

    scanf("%d",&size);

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

    printf("Enter Text...\n");

    fflush(stdin);

    fgets(p,size,stdin);

    p[strlen(p)-1] = '\0';

    printf("Entered Text...\n");

    printf("%s\n",p);

    free(p);

    return 0;

}

1. Write a program to read a one dimensional array, print sum of all elements along with inputted array elements using dynamic memory allocation.

Ans-

// 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,sum=0;

    int \*p;

    printf("Enter Size of array\n");

    scanf("%d",&size);

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

    printf("Enter array elements\n");

    for(int i=0;i<size;i++){

        scanf("%d",p+i);

    }

    for(int i=0;i<size;i++){

        sum = sum + \*(p+i);

    }

    printf("Sum of Elements: %d",sum);

    free(p);

    return 0;

}

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

Ans-

// 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,largest;

    int \*p;

    printf("Enter Size of array\n");

    scanf("%d",&size);

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

    printf("Enter array elements\n");

    for(int i=0;i<size;i++){

        scanf("%d",p+i);

    }

    // finding largest element

    largest = -1;

    for(int j=0;j<size;j++){

        if(\*(p+j)>largest)

            largest = \*(p+j);

    }

    printf("Largest Element: %d\n",largest);

    free(p);

    return 0;

}

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

Ans-

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

#include<stdio.h>

#include<stdlib.h>

int main(){

    int \*p,x=5;

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

    p = &x; // Memory leak(pointer is updated without free)

    return 0;

}

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

Ans-

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

#include<stdio.h>

int \* fun(){

    int x;

    return &x;

}

int main(){

    int \*p;

    p = fun(); //Dangling pointer

    return 0;

}

1. 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.

Ans-

// 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;

    int \*p;

    printf("Enter size in bytes to allocate\n");

    scanf("%d",&size);

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

    if(p == NULL){

        printf("Memory allocation Failed\n");

        return -1;

    }

    printf("Memory Allocated Successfully\n");

    free(p);

    return 0;

}

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

Ans-

// 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 size,max,min;

    int \*p;

    printf("Enter Size of array\n");

    scanf("%d",&size);

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

    printf("Enter array elements\n");

    for(int i=0;i<size;i++){

        scanf("%d",p+i);

    }

    // finding max element

    max = -1;

    for(int j=0;j<size;j++){

        if(\*(p+j)>max)

            max = \*(p+j);

    }

    // finding min element

    min = \*p;

    for(int j=0;j<size;j++){

        if(\*(p+j)<min)

            min = \*(p+j);

    }

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

    free(p);

    return 0;

}