

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>
char *input();
int main()
{
    char *ptr;
    ptr = input();
    printf("String is : %s", ptr);
    free(ptr);
    return 0;
}

char *input()
{
    char *p, c;
    int i = 0, j = 1;
    p = (char *)malloc(sizeof(char));
    printf("Enter any string : ");
    while (c != '\n')
    {
        c = getc(stdin);
        j++;
        p = (char *)realloc(p, j * sizeof(char));
        p[i] = c;
        i++;
    }
    p[i] = '\0';
    return p;
}

=====
Output:
Enter any string : Akhtar
String is : Akhtar
```

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 *ptr, i, n;
    float sum = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &n);

    ptr = (int *)calloc(n, sizeof(int));
    if (ptr == NULL)
    {
        printf("\nMemory Allocation Failed");
        return 0;
    }

    printf("Enter %d numbers : ", n);
    for (i = 0; i < n; i++)
        scanf("%d", ptr+i);
```

```

    for (i = 0; i < n; i++)
        sum = sum + (*(ptr+i));

    printf("Average of data values : %.2f", sum/n);
    free(ptr);

    return 0;
}

```

Output:

```

Enter number of element to store in array : 10
Enter 10 numbers : 10 20 30 40 50 60 70 80 90 111
Average of data values : 56.10

```

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 *p, i, size, sum = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &size);

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

    printf("Enter %d numbers : ", size);
    for (i = 0; i < size; i++)
        scanf("%d", p+i);

    for(i=0;i<size;i++)
        sum = sum + *(p+i);

    printf("Sum of %d numbers are %d", size, sum);
    free(p);

    return 0;
}

```

Output:

```

Enter number of element to store in array : 6
Enter 6 numbers : 56 25 87 96 31 85
Sum of 6 numbers are 380

```

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

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    char *p, c;
    int i = 0, n = 0;
    p = (char *)malloc(sizeof(char));
    printf("Enter any string : ");
    while (c != '\n')
    {
        c = getc(stdin);
        n++;
        p = (char *)realloc(p, n * sizeof(char));
        p[i] = c;
        i++;
    }
    p[i - 1] = '\0';
}

```

```

    printf("String is \"%s\\n\"", p);
    free(p);
    return 0;
}

```

Output:

```

Enter any string : Akhtar
String is "Akhtar"

```

- 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 *p, i, size, sum = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &size);

    p = (int *)calloc(size, sizeof(int));
    if (p == NULL)
    {
        printf("\nMemory Allocation Failed");
        return 0;
    }

    printf("Enter %d numbers : ", size);
    for (i = 0; i < size; i++)
        scanf("%d", p+i);

    for(i=0;i<size;i++)
        sum = sum + (*(p+i));

    printf("Sum of %d numbers are %d", size, sum);
    free(p);

    return 0;
}

```

Output:

```

Enter number of element to store in array : 8
Enter 8 numbers : 3 5 9 7 8 12 58 23
Sum of 8 numbers are 125

```

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

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *p, i, size, sum = 0, temp = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &size);

    p = (int *)calloc(size, sizeof(int));
    if (p == NULL)
    {
        printf("\nMemory Allocation Failed");
        return 0;
    }

```

```

printf("Enter %d numbers : ", size);
for (i = 0; i < size; i++)
    scanf("%d", p + i);

for (i = 0; i < size - 1; i++)
{
    for (int j = i + 1; j < size; j++)
    {
        if (*(p + i) > *(p + j))
        {
            temp = *(p + i);
            *(p + i) = *(p + j);
            *(p + j) = temp;
        }
    }
}

printf("The largest element is %d", *(p+size-1));
free(p);

return 0;
}

```

Output:

```

Enter number of element to store in array : 10
Enter 10 numbers : 10 20 30 40 50 60 70 80 90 10
The largest element is 90

```

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

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    int *p, sum = 0;
    p = (int *)calloc(2, sizeof(int));
    printf("Enter two numbers : ");
    scanf("%d%d", p+0, p+1);
    sum = *(p+0) + *(p+1);
    printf("Sum is %d", sum);
    p = NULL;
    if (p == NULL)
    {
        printf("\nMemory leak");
        return 0;
    }

    return 0;
}

```

Output:

```

Enter two numbers : 10 20
Sum is 30
Memory leak

```

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

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *p;
    p = (int *)calloc(1, sizeof(int));
    printf("Enter a number : ");
    scanf("%d", p);
}

```

```

printf("Before free %d", *p);
free(p);
printf("\nAfter free %d", *p);
return 0;
}

```

Output:

```

Enter a number : 20
Before free 20
After free -137939136

```

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 *ptr, i, n;
    float sum = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &n);

    ptr = (int *)calloc(n, sizeof(int));
    if (ptr == NULL)
    {
        printf("\nMemory Allocation Failed");
        return 0;
    }

    printf("Enter %d numbers : ", n);
    for (i = 0; i < n; i++)
        scanf("%d", ptr+i);

    for (i = 0; i < n; i++)
        sum = sum + (*(ptr+i));

    printf("Average of data values : %.2f", sum/n);
    free(ptr);

    return 0;
}

```

Output:

```

Enter number of element to store in array : 5
Enter 5 numbers : 10 20 30 40 50
Average of data values : 30.00

```

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 *p, i, size, sum = 0, temp = 0;

    printf("Enter number of element to store in array : ");
    scanf("%d", &size);

    p = (int *)calloc(size, sizeof(int));
    if (p == NULL)
    {
        printf("\nMemory Allocation Failed");
    }
}

```

```

        return 0;
    }

    printf("Enter %d numbers : ", size);
    for (i = 0; i < size; i++)
        scanf("%d", p + i);

    for (i = 0; i < size - 1; i++)
    {
        for (int j = i + 1; j < size; j++)
        {
            if (*(p + i) > *(p + j))
            {
                temp = *(p + i);
                *(p + i) = *(p + j);
                *(p + j) = temp;
            }
        }
    }

    printf("Minimum is %d", *p);
    printf("\nMaximum is %d", *(p + size - 1));
    free(p);

    return 0;
}

```

=====

Output:

```

Enter number of element to store in array : 8
Enter 8 numbers : 50 60 80 100 30 40 98 32
Minimum is 30
Maximum is 100

```