

# Task 1

Description:

Creating a simple calculator in C programming using the  
If selection statement,  
switch Multiple selection statement,  
while iteration Statement,  
Counter Control iteration,

This program takes an arithmetic operator +, -, \*, / and two operands from the user. Then, it performs the calculation on the two operands depending upon the operator entered by the user.

The \* operator entered by the user is stored in op. And, the two operands, 1.5 and 4.5 are stored in first and second respectively.

Since the operator \* matches case '\*', the control of the program jumps to

```
printf("%.1f * %.1f = %.1f", first, second, first * second);
```

This statement calculates the product and displays it on the screen.

To make our output look cleaner, we have simply limited the output to one decimal place using the code %.1f.

Finally, the break; statement ends the switch statement

**Code:**

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char op;
7      int i = 0;
8      double first, second;
9      while (op != 'q')
10     {
11         printf("\n\t\tEnter an operator (+, -, *, /)(q/ Quit): ");
12         if (i > 0)
13         {
14             char ch = getchar();
15         }
16         i++;
17         scanf("%c", &op);
18         if (op == 'q')
19         {
20             printf("Calculator is Terminating\n");
21             return 0;
22         }
23         printf("\nTwo operands: ");
24         printf("\nEnter first operands: ");
25         scanf("%lf", &first);
26         printf("Enter second operands: ");
27         scanf("%lf", &second);
28
29         switch (op)
30         {
31             case '+':
32                 printf("%.11f + %.11f = %.11f", first, second, first + second);
33                 break;
34             case '-':
35                 printf("%.11f - %.11f = %.11f", first, second, first - second);
36                 break;
37             case '*':
38                 printf("%.11f * %.11f = %.11f", first, second, first * second);
39                 break;

```

```

39                 break;
40             case '/':
41                 printf("%.11f / %.11f = %.11f", first, second, first / second);
42                 break;
43             // operator doesn't match any case constant
44             default:
45                 printf("Error! operator is not correct");
46         }
47     }
48     return 0;
49 }
50

```

# Output:

```
Enter an operator (+, -, *, /)(q/ Quit): +  
  
Two operands:  
Enter first operands: 2  
Enter second operands: 2  
2.0 + 2.0 = 4.0  
Enter an operator (+, -, *, /)(q/ Quit): *  
  
Two operands:  
Enter first operands: 3  
Enter second operands: 4  
3.0 * 4.0 = 12.0  
Enter an operator (+, -, *, /)(q/ Quit): -  
  
Two operands:  
Enter first operands: 40  
Enter second operands: 10  
40.0 - 10.0 = 30.0  
Enter an operator (+, -, *, /)(q/ Quit): /  
  
Two operands:  
Enter first operands: 20  
Enter second operands: 2  
20.0 / 2.0 = 10.0  
Enter an operator (+, -, *, /)(q/ Quit): q  
Calculator is Terminating
```

## Task 2

### Number Converter:

Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

If else... selection statement,

Nested loop Counter statement

switch Multiple selection statement,

while iteration Statement,

Counter Control iteration,

For iteration statement,

Here is a program for numbers conversion. This program can convert Binary number, decimal number and hexadecimal number to each other respectively.

By entering choice, you can get choice for number conversion.

If user can choice 1, it will convert binary to decimal

For choice 1, it will take binary number and it will output will be in decimal number.

If user can choice 2, it will convert binary to hexa-decimal

For choice 2, it will take binary number and it will output will be in hexa-decimal number.

If user can choice 3, it will convert decimal to binary

For choice 3, it will take decimal number and it will output will be in binary number.

If user can choice 4, it will convert decimal to hexa-decimal

For choice 4, it will take decimal number and it will output will be in hexa-decimal number.

If user can choice 5, it will convert hexa-decimal to binary

For choice 5, it will take hexa-decimal number and it will output will be in binary number.

If user can choice 6, it will convert hexa-decimal to decimal

For choice 6, it will take hexa-decimal number and it will output will be in decimal number.

**Code:**

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <string.h>
4  #include <conio.h>
5
6  long int Binary_to_Decimal(long int);
7  long int Binary_to_Hexadecimal(long int);
8  long int Decimal_to_Binary(long int);
9  long int Decimal_to_Hexadecimal(long int);
10 void Hexadecimal_to_Binary(char[]);
11 void Hexadecimal_to_Decimal(char[]);
12
13 int main()
14 {
15     int operations, numbers = 1, checking;
16     long int b, o, d;
17     char h[100];
18     int x, y, spacebar;
19
20     printf("\t\tWELCOME TO NUMBER SYSTEM CONVERSION\n\n");
21
22     while (numbers != 0)
23     {
24         printf("\t\t>>>>>> CHOOSE THE CONVERSION <<<<<<\n\n");
25
26         printf("\n=> BINARY <=\n");
27         printf("1: Binary to Decimal.\n2: Binary to Hexa-Decimal.\n");
28
29         printf("\n=> DECIMAL <=\n");
30         printf("3: Decimal to Binary.\n4: Decimal to Hexa-Decimal.\n");
31
32         printf("\n=> HEXA-DECIMAL <=\n");
33         printf("5: Hexa-Decimal to Binary.\n6: Hexa-Decimal to Decimal.\n");
34
35         printf("\nENTER YOUR CHOICE: ");
36         scanf("%d", &operations);
37
38         switch (operations)
39         {
40             case 1:

```

```

40     case 1:
41         printf("\n***BINARY TO DECIMAL***\n");
42     D:
43         printf("\nEnter the Number in Binary form (0s & 1s): ");
44         scanf("%ld", &b);
45
46         checking = b;
47
48         while (checking != 0)
49         {
50             numbers = checking % 10;
51             if (numbers > 1)
52             {
53                 printf("\n%d IS NOT BINARY NUMBER.\n", b);
54                 printf("***TRY AGAIN***\n");
55                 goto D;
56             }
57             else
58                 checking = checking / 10;
59         }
60
61         Binary_to_Decimal(b);
62         break;
63
64     case 2:
65         printf("\n***BINARY TO HEXA-DECIMAL***\n");
66     F:
67         printf("\nEnter the Number in Binary form (0s & 1s): ");
68         scanf("%ld", &b);
69
70         checking = b;
71
72         while (checking != 0)
73         {
74             numbers = checking % 10;
75             if (numbers > 1)
76             {
77                 printf("\n%d IS NOT BINARY NUMBER.\n", b);
78                 printf("***TRY AGAIN***\n");
79                 goto F;

```

```

80     }
81     else
82     |     checking = checking / 10;
83     }
84
85     Binary_to_Hexadecimal(b);
86     break;
87
88     case 3:
89         printf("\n***DECIMAL TO BINARY***\n");
90         printf("\nEnter the Number in Decimal form (0 to 9): ");
91         scanf("%ld", &d);
92         Decimal_to_Binary(d);
93         break;
94
95     case 4:
96         printf("\n***DECIMAL TO HEXA-DECIMAL***\n");
97         printf("\nEnter the Number in Decimal form (0 to 9): ");
98         scanf("%ld", &d);
99         Decimal_to_Hexadecimal(d);
100
101         break;
102
103     case 5:
104         printf("\n***HEXA-DECIMAL TO BINARY***\n");
105     X:
106         printf("\nEnter the Number in Hexa-Decimal form: ");
107         scanf("%s", &h);
108         //check
109         for (x = strlen(h) - 1; x >= 0; x--)
110         {
111             if (h[x] > 'f' && h[x] <= 'z' || h[x] > 'F' && h[x] <= 'Z')
112             {
113                 printf("\nYou have to Enter Hexa-Decimal Number.\n");
114                 printf("%c IS NOT Hexa-Decimal Number.\n", h[x]);
115                 goto X;
116             }
117         }
118         Hexadecimal_to_Binary(h);
119         break;

```



```

120
121     case 6:
122         printf("\n***HEXA-DECIMAL TO DECIMAL***\n");
123     Y:
124         printf("\nEnter the Number in Hexa-Decimal form: ");
125         scanf("%s", &h);
126         //check
127         for (x = strlen(h) - 1; x >= 0; x--)
128         {
129             if (h[x] > 'f' && h[x] <= 'z' || h[x] > 'F' && h[x] <= 'Z')
130             {
131                 printf("\nYou have to Enter Hexa-Decimal Number.\n");
132                 printf("'%c' IS NOT Hexa-Decimal Number.\n", h[x]);
133                 goto Y;
134             }
135         }
136         Hexadecimal_to_Decimal(h);
137
138         break;
139
140     default:
141         printf("\n***INVALID NUMBER***\n");
142         break;
143     }
144     printf("\n\nDO YOU WANT TO CONTINUE = (1/0) :\n");
145     scanf("%d", &numbers);
146 }
147
148 spacebar = 3 + 35;
149 for (x = 1; x <= 3; x++)
150 {
151     for (y = 1; y <= spacebar; y++)
152     {
153         printf(" ");
154     }
155     spacebar--;
156     for (y = 1; y <= 2 * x - 1; y++)
157     {
158         printf("*");
159     }

```

```

160     printf("\n");
161 }
162 spacebar = 37;
163 for (x = 1; x <= 3; x++)
164 {
165     for (y = 1; y <= spacebar; y++)
166     {
167         printf(" ");
168     }
169     spacebar++;
170     for (y = 1; y <= 2 * (3 - x) - 1; y++)
171     {
172         printf("*");
173     }
174
175     printf("\n");
176 }
177 printf("\t\t\t BROUGHT TO YOU BY ITSOURCECODE.COM \n\t\t\t Log On now For More C/C++ Free Projects");
178 }
179
180 long int Binary_to_Decimal(long int bin)
181 {
182     int remainder, summation = 0, x = 0;
183     while (bin != 0)
184     {
185         remainder = bin % 10;
186         bin = bin / 10;
187         summation = summation + remainder * pow(2, x);
188         x++;
189     }
190
191     printf("\nEquivalent Decimal Number : %d", summation);
192 }
193
194 long int Binary_to_Hexadecimal(long int bin)
195 {
196     int remainder, x = 0, summation = 0, remaining[100], length = 0;
197
198     while (bin != 0)
199     {

```

```

200     remainder = bin % 10;
201     bin = bin / 10;
202     summation = summation + remainder * pow(2, x);
203     x++;
204 }
205 x = 0;
206 while (summation != 0)
207 {
208     remaining[x] = summation % 16;
209     summation = summation / 16;
210     x++;
211     length++;
212 }
213 printf("\nEquivalent Hexa-Decimal Number : ");
214 for (x = length - 1; x >= 0; x--)
215 {
216     switch (remaining[x])
217     {
218     case 10:
219         printf("A");
220         break;
221
222     case 11:
223         printf("B");
224         break;
225
226     case 12:
227         printf("C");
228         break;
229
230     case 13:
231         printf("D");
232         break;
233
234     case 14:
235         printf("E");
236         break;
237
238     case 15:
239         printf("F");
240         break;
241
242     default:
243         printf("%d", remaining[x]);
244     }
245 }
246 }
247
248 long int Decimal_to_Binary(long int dec)

```

```

249 {
250     int remainder[50], x, length = 0;
251     do
252     {
253         remainder[x] = dec % 2;
254         dec = dec / 2;
255         x++;
256         length++;
257     } while (dec != 0);
258
259     printf("\nEquivalent Binary Number : ");
260     for (x = length - 1; x >= 0; x--)
261     {
262         printf("%d", remainder[x]);
263     }
264 }
265
266 long int Decimal_to_Hexadecimal(long int dec)
267 {
268     int remainder[50], x, length = 0;
269     do
270     {
271         remainder[x] = dec % 16;
272         dec = dec / 16;
273         x++;
274         length++;
275     } while (dec != 0);
276
277     printf("\nEquivalent Hexa-Decimal Number : ");
278     for (x = length - 1; x >= 0; x--)
279     {
280         switch (remainder[x])
281         {
282             case 10:
283                 printf("A");
284                 break;
285
286             case 11:
287                 printf("B");
288                 break;
289
290             case 12:
291                 printf("C");
292                 break;
293
294             case 13:
295                 printf("D");
296                 break;
297

```

```

298     case 14:
299         printf("E");
300         break;
301
302     case 15:
303         printf("F");
304         break;
305
306     default:
307         printf("%d", remainder[x]);
308     }
309 }
310 }
311
312 void Hexadecimal_to_Binary(char hex[])
313 {
314     int x = 0;
315     printf("\nEquivalent Binary Number : ");
316     for (x = 0; x < strlen(hex); x++)
317     {
318         switch (hex[x])
319         {
320             case '0':
321                 printf("0000");
322                 break;
323             case '1':
324                 printf("0001");
325                 break;
326             case '2':
327                 printf("0010");
328                 break;
329             case '3':
330                 printf("0011");
331                 break;
332             case '4':
333                 printf("0100");
334                 break;
335             case '5':
336                 printf("0101");
337                 break;
338             case '6':
339                 printf("0110");
340                 break;
341             case '7':
342                 printf("0111");
343                 break;
344             case '8':
345                 printf("1000");
346                 break;

```

```

347     case '9':
348         printf("1001");
349         break;
350     case 'A':
351     case 'a':
352         printf("1010");
353         break;
354     case 'B':
355     case 'b':
356         printf("1011");
357         break;
358     case 'C':
359     case 'c':
360         printf("1100");
361         break;
362     case 'D':
363     case 'd':
364         printf("1101");
365         break;
366     case 'E':
367     case 'e':
368         printf("1110");
369         break;
370     case 'F':
371     case 'f':
372         printf("1111");
373         break;
374
375     default:
376         printf("\n Invalid hexa digit %c ", hex[x]);
377     }
378 }
379 }
380
381 void Hexadecimal_to_Decimal(char hex[])
382 {
383     int x, numbers = 0, powered = 0, decimal = 0;
384
385     for (x = strlen(hex) - 1; x >= 0; x--)
386     {
387         if (hex[x] == 'A' || hex[x] == 'a')
388         {
389             numbers = 10;
390         }
391         else if (hex[x] == 'B' || hex[x] == 'b')
392         {
393             numbers = 11;
394         }
395         else if (hex[x] == 'C' || hex[x] == 'c')

```

```
395     else if (hex[x] == 'C' || hex[x] == 'c')
396     {
397         numbers = 12;
398     }
399     else if (hex[x] == 'D' || hex[x] == 'd')
400     {
401         numbers = 13;
402     }
403     else if (hex[x] == 'E' || hex[x] == 'e')
404     {
405         numbers = 14;
406     }
407     else if (hex[x] == 'F' || hex[x] == 'f')
408     {
409         numbers = 15;
410     }
411     else
412         //(a[i]>=0 || a[i]<=9)
413     {
414         numbers = hex[x] - 48;
415     }
416
417     decimal = decimal + numbers * pow(16, powered);
418     powered++;
419 }
420 printf("\nEquivalent Decimal Number : %d", decimal);
421 }
```

**Output:**

WELCOME TO NUMBER SYSTEM CONVERSION

>>>>> CHOOSE THE CONVERSION <<<<<<

=> BINARY <=  
1: Binary to Decimal.  
2: Binary to Hexa-Decimal.

=> DECIMAL <=  
3: Decimal to Binary.  
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=  
5: Hexa-Decimal to Binary.  
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 1

\*\*\*BINARY TO DECIMAL\*\*\*

Enter the Number in Binary form (0s & 1s): 1010111

Equivalent Decimal Number : 87

DO YOU WANT TO CONTINUE = (1/0) :  
1

>>>>> CHOOSE THE CONVERSION <<<<<<

=> BINARY <=  
1: Binary to Decimal.  
2: Binary to Hexa-Decimal.

=> DECIMAL <=  
3: Decimal to Binary.  
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=  
5: Hexa-Decimal to Binary.  
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 2

\*\*\*BINARY TO HEXA-DECIMAL\*\*\*

Enter the Number in Binary form (0s & 1s): 101011

Equivalent Hexa-Decimal Number : 2B

DO YOU WANT TO CONTINUE = (1/0) :  
1

>>>>> CHOOSE THE CONVERSION <<<<<<



```
=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.

=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 4

***DECIMAL TO HEXA-DECIMAL***

Enter the Number in Decimal form (0 to 9): 12

Equivalent Hexa-Decimal Number : C

DO YOU WANT TO CONTINUE = (1/0) :
1

>>>>> CHOOSE THE CONVERSION <<<<<<
```

```
=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.

=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 5

***HEXA-DECIMAL TO BINARY***

Enter the Number in Hexa-Decimal form: AC01

Equivalent Binary Number : 1010110000000001

DO YOU WANT TO CONTINUE = (1/0) :
1

>>>>> CHOOSE THE CONVERSION <<<<<<
```

```
=> BINARY <=
1: Binary to Decimal.
2: Binary to Hexa-Decimal.

=> DECIMAL <=
3: Decimal to Binary.
4: Decimal to Hexa-Decimal.

=> HEXA-DECIMAL <=
5: Hexa-Decimal to Binary.
6: Hexa-Decimal to Decimal.

ENTER YOUR CHOICE: 6

***HEXA-DECIMAL TO DECIMAL***

Enter the Number in Hexa-Decimal form: AC

Equivalent Decimal Number : 172

DO YOU WANT TO CONTINUE = (1/0) :
█
```

### TASK 3

Adding, Deleting, Search and Display numbers

Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

If else... selection statement,

switch Multiple selection statement,

while iteration Statement,

Counter Control iteration,

For iteration statement,

In this program we can add delete search and display items.

There is choice option for each.

If user Enter 0, it will quit program.

If user Enter 1, it will ask for number and add a number.

#### **Inserting An Element in An Array**

Inserting an element in an unsorted array is faster as compared to sorted array. This is because in an unsorted array, you do not have to worry about the new position of the element. The position of the new element is the last position in the array.

If user Enter 2, it will ask for number and delete a number.

#### **Deleting An Element from An Array**

To delete an element in an array, we need to first search the element. Then we need to delete the element and shift the rest of the elements to the left

If user Enter 3, it will ask for number and search a number.

### **Searching an Element in an Array**

To search an element in an array you need to traverse through the array using loop and search for the given element.

If user Enter 4, it will display all numbers.

**Code:**

```

1  #include <stdio.h>
2
3  int findElement(int array[], int size, int keyToBeSearched)
4  {
5      int i;
6      // Finding & returning the position of the element
7      for (i = 0; i < size; i++)
8          if (array[i] == keyToBeSearched)
9              return i;
10     return -1;
11 }
12
13 //Function to insert an element in an array.
14 //arr[] = array, elements = number of elements present in the array
15 //keyToBeInserted = element to be inserted in the array
16 // size of the array
17 int insertElement(int arr[], int elements, int keyToBeInserted, int size)
18 {
19     // Check if the capacity of the array is already full
20     if (elements >= size)
21         return elements;
22     //If not then the element is inserted at the last index
23     //and the new array size is returned
24     arr[elements] = keyToBeInserted;
25     return (elements + 1);
26 }
27
28 // Function to delete an element
29 // where array[] is the array from which element needs to be deleted
30 // size is the size of the array
31 // keyToBeDeleted is the element to be deleted from the array
32 int deleteElement(int array[], int size, int keyToBeDeleted)
33 {
34     // Calling findElement function to get the position of the element which needs to be deleted
35     int pos = findElement(array, size, keyToBeDeleted);
36     // If element is not found then it prints Element not found
37     if (pos == -1)
38     {
39         printf("Element not found");
40         return size;
41     }
42     // Otherwise it deletes the element & moves rest of the element by one position
43     int i;
44     for (i = pos; i < size - 1; i++)
45         array[i] = array[i + 1];
46     return size - 1;
47 }
48
49 void displayArray(int array[], int size)

```

```

48
49 void displayArray(int array[], int size)
50 {
51     if (size <= 0)
52     {
53         printf("Element not found");
54         return;
55     }
56     // Otherwise it deletes the element & moves rest of the element by one position
57     int i;
58     printf("{");
59     for (i = 0; i < size; i++)
60     {
61         printf("%d,", array[i]);
62     }
63     printf("}\n");
64 }
65
66 int main()
67 {
68     int array[20] = {0};
69     int size = sizeof(array) / sizeof(array[0]);
70     int eleCount = 0;
71
72     int num3, num2, num;
73
74     int run = 0;
75     int choice;
76
77     while (choice != 0)
78     {
79         printf("\nEnter 0. Quit\n");
80         printf("Enter 1. Add Number\n");
81         printf("Enter 2. Delete Number\n");
82         printf("Enter 3. Search Number\n");
83         printf("Enter 4. Display Numbers\n");
84         printf("Enter Choice: ");
85         scanf("%d", &choice);
86
87         switch (choice)
88         {
89             case 0:
90                 printf("Quiting ");
91                 return 0;
92                 break;
93             case 1:
94                 printf("ADDING NUMBER\n");
95                 printf("Enter a number : ");
96                 scanf("%d", &num);

```

```

94     printf("Adding Number\n");
95     printf("Enter a number : ");
96     scanf("%d", &num);
97     eleCount = insertElement(array, eleCount, num, size);
98
99     break;
100 case 2:
101     printf("DELETING NUMBER\n");
102     printf("Enter a number : ");
103     scanf("%d", &num2);
104     eleCount = deleteElement(array, eleCount, num2);
105
106     break;
107 case 3:
108     printf("SEARCHING NUMBER\n");
109     printf("Enter a number : ");
110     scanf("%d", &num3);
111     int index = findElement(array, size, num3);
112     printf("Index of %d is : %d\n", num3, index);
113     break;
114 case 4:
115     printf("DISPLAY NUMBERS\n");
116     displayArray(array, eleCount);
117     break;
118
119 default:
120     printf("Invalid Choice \n");
121     break;
122 }
123 }
124
125 return 0;
126 }

```

Output:

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 1
ADDING NUMBER
Enter a number : 2
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 1
ADDING NUMBER
Enter a number : 3
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 1
ADDING NUMBER
Enter a number : 5
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 1
ADDING NUMBER
Enter a number : 7
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 2
DELETING NUMBER
Enter a number : 5
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 1
```

```
ADDING NUMBER
Enter a number : 7
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 2
```

```
DELETING NUMBER
Enter a number : 5
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 4
```

```
DISPLAY NUMBERS
{2,3,7,}
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 3
```

```
SEARCHING NUMBER
Enter a number : 7
Index of 7 is : 2
```

```
Enter 0. Quit
Enter 1. Add Number
Enter 2. Delete Number
Enter 3. Search Number
Enter 4. Display Numbers
Enter Choice: 0
```

```
Quiting
```



# Task4

## Matrix

### Description:

The topics we covered in this program is:

Control Structure,

If selection statement,

If else... selection statement,

Counter Control iteration,

For iteration statement,

Arrays in C language

In this program 2D array is used,

As program starts it take input from user, it will fill matrix with entries,

```
void fillMatrix (int arr[N][N])
```

it will fill the matrix

```
void displayMatrix (int arr[N][N]);
```

then it would display matrix on console.

```
bool isDiagonalMatrix(int mat[N][N]);
```

it will check whether matrix is diagonal or not and return true or false.

Matrix is diagonal if all the entries except diagonals are zero.

```
bool isScalarMatrix (int mat[N][N]);
```

it will check whether matrix is scalar or not and return true or false.

```
bool isUpperTriangularMatrix (int mat[N][N]);
```

it will check whether matrix is upper triangular or not and return true or false.

Matrix is upper triangular if all the entries of diagonal and upper entries of diagonal are non-zero.

```
bool check_lower_triangular_matrix (int mat[N][N]);
```

it will check whether matrix is lower triangular or not and return true or false.

Matrix is lower triangular if all the entries of diagonal and lower entries of diagonal are non-zero.

```
void transpose (int arr[N][N], int brr[N][N])
```

It will tale transpose of the matrix.

In transpose the entries in the rows change to columns and columns to rows respectively.

In this program we can add delete search and display items.

There is choice option for each.

If user Enter 0, it will quit program.

If user Enter 1, it will ask for number and add a number.

### **Inserting An Element in An Array**

Inserting an element in an unsorted array is faster as compared to sorted array. This is because in an unsorted array, you do not have to worry about the new position of the element. The position of the new element is the last position in the array.

If user Enter 2, it will ask for number and delete a number.

### **Deleting An Element from An Array**

To delete an element in an array, we need to first search the element. Then we need to delete the element and shift the rest of the elements to the left

If user Enter 3, it will ask for number and search a number.

### **Searching an Element in an Array**

To search an element in an array you need to traverse through the array using loop and search for the given element.

If user Enter 4, it will display all numbers.

```
1  #include <stdio.h>
2  #include <stdbool.h>
3  #define N 3
4
5  void fillMatrix(int arr[N][N])
6  {
7      int i = 0, j;
8      for (i = 0; i < N; i++)
9      {
10         for (j = 0; j < N; j++)
11         {
12             printf("Enter [%d][%d]: ", i, j);
13             scanf("%d", &arr[i][j]);
14         }
15     }
16 }
17
18 void displayMatrix(int arr[N][N])
19 {
20     printf("Matrix[R][C]\n");
21     int i = 0, j;
22     for (i = 0; i < N; i++)
23     {
24         for (j = 0; j < N; j++)
25         {
26             printf("%d\t", arr[i][j]);
27         }
28         printf("\n\n");
29     }
30 }
31
32 bool isDiagonalMatrix(int mat[N][N])
33 {
34     for (int i = 0; i < N; i++)
35     for (int j = 0; j < N; j++)
36     {
37         // condition to check other elements
38         // except main diagonal are zero or not.
39         if ((i != j) && (mat[i][j] != 0))
40             return false;
```

Code:

```

40         return false;
41     return true;
42 }
43
44 bool isScalarMatrix(int mat[N][N])
45 {
46     // Check all elements except main diagonal are
47     // zero or not.
48     for (int i = 0; i < N; i++)
49         for (int j = 0; j < N; j++)
50             if ((i != j) && (mat[i][j] != 0))
51                 return false;
52
53     // Check all diagonal elements are same or not.
54     for (int i = 0; i < N - 1; i++)
55         if (mat[i][i] != mat[i + 1][i + 1])
56             return false;
57     return true;
58 }
59
60 bool isUpperTriangularMatrix(int mat[N][N])
61 {
62     for (int i = 1; i < N; i++)
63         for (int j = 0; j < i; j++)
64             if (mat[i][j] != 0)
65                 return false;
66     return true;
67 }
68
69 bool check_lower_triangular_matrix(int mat[N][N])
70 {
71     int i, j;
72     for (i = 0; i < N; i++)
73         for (j = i + 1; j < N; j++)
74             if (mat[i][j] != 0)
75                 return false;
76     return true;
77 }
78

```

```
void transpose(int arr[N][N], int brr[N][N]) //Function Definition
{
    for (int i = 0; i < N; i++) //Transpose Matrix initialization
    {
        for (int j = 0; j < N; j++)
        {
            brr[j][i] = arr[i][j]; //Store elements in the transpose matrix
        }
    }
    printf("\nAfter transpose the elements are...\n");
    for (int i = 0; i < N; i++) //Print the transpose matrix
    {
        for (int j = 0; j < N; j++)
        {
            printf("%d\t", brr[i][j]);
        }
        printf("\n\n");
    }
}
```

```

98
99 int main()
100 {
101     /* 2D array declaration*/
102     int disp[N][N];
103
104     /*Counter variables for the loop*/
105
106     fillMatrix(disp);
107     displayMatrix(disp);
108     if (isDiagonalMatrix(disp))
109     {
110         printf("Matrix is Diagonal \n");
111     }
112     else
113     {
114         printf("Matrix is not Diagonal\n");
115     }
116     if (isUpperTriangularMatrix(disp))
117     {
118         printf("Matrix is Upper Triangular Matrix\n");
119     }
120     else
121     {
122         printf("Matrix is Not Upper Triangular Matrix\n");
123     }
124     if (check_lower_triangular_matrix(disp))
125     {
126         printf("Matrix is Lower Triangular Matrix\n");
127     }
128     else
129     {
130         printf("Matrix is Not Lower Triangular Matrix\n");
131     }
132     if (isScalarMatrix(disp))
133     {
134         printf("Matrix is Scalar \n");
135     }
136     else
137     {
138         printf("Matrix is not Scalar\n");
139     }
140     int tran[N][N];
141     transpose(disp, tran);
142
143     return 0;
144 }

```

## OUTPUT:

```
Enter [0][0]: 1
Enter [0][1]: 2
Enter [0][2]: 3
Enter [1][0]: 0
Enter [1][1]: 4
Enter [1][2]: 0
Enter [2][0]: 0
Enter [2][1]: 0
Enter [2][2]: 5
Matrix[R][C]
1      2      3

0      4      0

0      0      5

Matix is not Diagonal
Matrix is Upper Triangular Matix
Matrix is Not Lower Triangular Matix
Matix is not Scalar

After transpose the elements are...
1      0      0

2      4      0

3      0      5
```