



1) Write a C program to do the calculator's basic operations using pointers. Also terminate the execution only based on user's choice.

**Code :**

```
1  #include<stdio.h>
2  #include<conio.h>
3  #define FLUSH while(getchar()!='\n');
4
5  int main() {
6      int num1,num2;
7      int result;
8      char ch;
9      int flag=1;
10
11     while(flag==1) {
12
13         printf("\nEnter the 1st number : ");
14         scanf("%d",&num1);
15         printf("Enter the 2nd number : ");
16         scanf("%d",&num2);
17
18         FLUSH;
19
20         printf("Enter your choice : ");
21         scanf("%c",&ch);
22
23         int *p;
24         p=&num1;
25
26         int *q;
27         q=&num2;
28
29         switch(ch){
30             case '+':
31                 result = *p + *q;
```

```

32     printf("The result is : %d", result);
33     break;
34
35     case '-':
36         result = *p - *q;
37         printf("The result is : %d", result);
38         break;
39
40     case '*':
41         result = *p * *q;
42         printf("The result is : %d", result);
43         break;
44
45     case '/':
46         result = *p / *q;
47         printf("The result is : %d", result);
48         break;
49
50     default:
51         printf("Invalid Operation");
52 }
53
54 printf("\nEnter 1 to continue : ");
55 scanf("%d",&flag);
56 printf("\n");
57 }
58 return 0;
59 }
60

```

### Output :

```

Enter the 1st number : 7
Enter the 2nd number : 5
Enter your choice : *
The result is : 35
Enter 1 to continue : 1

Enter the 1st number : 5
Enter the 2nd number : 3
Enter your choice : -
The result is : 2
Enter 1 to continue : 0

Process returned 0 (0x0)   execution time : 29.006 s
Press any key to continue.

```

2) Write a C program to display the result of a student with N marks according to the VIT grading system using pointer to an array. (Note : calculate the average mark and fix the grade)

**Code :**

```
1  #include<stdio.h>
2  #include<conio.h>
3  #define FLUSH while(getchar()!='\n');
4
5  void Grading(int average) {
6      printf(" and the grade is : ");
7      if (average>95) { printf("S-Grade\n"); }
8      else if (average>=90 && average<=95) { printf("A-Grade\n"); }
9      else if (average>=85 && average<90) { printf("B-Grade\n"); }
10     else if (average>=70 && average<85) { printf("C-Grade\n"); }
11     else if (average<70) { printf("F-Grade\n"); }
12 }
13
14 void average1(int arr1[],int N) {
15     int sum1 = 0;
16     int average;
17
18     int *ptr = arr1;
19
20     for(int i=0;i<N;i++) {
21         sum1 = sum1 + *ptr;
22         ptr++;
23     }
24     average = sum1/N;
25     printf("Average mark is : %d",average);
26     Grading(average);
27 }
28
29 void average2(int *ptr,int N) {
30     int sum1 = 0;
31     int average;
32 }
```

```

33     for(int i=0;i<N;i++) {
34         sum1 = sum1 + *(ptr + i);
35     }
36     average = sum1/N;
37     printf("Average mark is : %d",average);
38     Grading(average);
39 }
40
41 int main()
42 {
43     int N;
44     printf("Enter the number of subjects : ");
45     scanf("%d",&N);
46
47     int arr1[N];
48
49     printf("Enter the marks of individual subjects : ");
50     for(int i=0;i<N;i++) {
51         scanf("%d",&arr1[i]);
52     }
53
54     FLUSH;
55
56     average1(arr1,N);
57     printf("\n");
58     average2(arr1,N);
59     return 0;
60 }
61

```

### Output

```

Enter the number of subjects : 5
Enter the marks of individual subjects : 89 97 95 79 84
Average mark is : 88 and the grade is : B-Grade

Average mark is : 88 and the grade is : B-Grade

Process returned 0 (0x0)   execution time : 18.564 s
Press any key to continue.

```

3) Repeat the same task in Q2 for M students using pointer to 2D array.

**Code :**

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  void Grading(int average) {
5      printf(" and the grade is : ");
6
7      if (average>95) {
8          printf("S-Grade\n");
9      }
10
11     else if (average>=90 && average<=95) {
12         printf("A-Grade\n");
13     }
14
15     else if (average>=85 && average<90) {
16         printf("B-Grade\n");
17     }
18
19     else if (average>=70 && average<85) {
20         printf("C-Grade\n");
21     }
22
23     else if (average<70) {
24         printf("F-Grade\n");
25     }
26 }
27
28 int main() {
29
30     int student_count;
31     int subject_count;
32
33     printf("Enter the number of students : ");
34     scanf("%d",&student_count);
35
36     printf("Enter the number of subjects : ");
37     scanf("%d",&subject_count);
38
39     int arr1[student_count][subject_count];
40
41     for(int i=0;i<student_count;i++) {
42         printf("Enter the marks of student%d : ",i+1);
43         for(int j=0;j<subject_count;j++) {
```

```

44         scanf("%d",&arr1[i][j]);
45     }
46 }
47
48 int sum1=0;
49 int average = 0;
50
51 for(int i=0;i<student_count;i++) {
52     for(int j=0;j<subject_count;j++) {
53         sum1 = sum1 + (*(arr1+i)+j);
54     }
55     average = sum1/subject_count;
56     printf("Average mark is : %d",average);
57     Grading(average);
58     sum1=0;
59 }
60 return 0;
61 }

```

### Output :

```

Enter the number of students : 5
Enter the number of subjects : 6
Enter the marks of student1 : 89 76 32 69 90 74
Enter the marks of student2 : 99 76 42 68 65 97
Enter the marks of student3 : 93 78 65 48 77 92
Enter the marks of student4 : 87 56 43 19 78 94
Enter the marks of student5 : 67 53 92 94 68 88
Average mark is : 71 and the grade is : C-Grade
Average mark is : 74 and the grade is : C-Grade
Average mark is : 75 and the grade is : C-Grade
Average mark is : 62 and the grade is : F-Grade
Average mark is : 77 and the grade is : C-Grade

Process returned 0 (0x0)   execution time : 71.065 s
Press any key to continue.

```

4) Write a C program to get your name and your friends name as input. Compare both of the names character by character and display the similar characters using pointers. Also display the count of similar and dissimilar characters.

#### Code

```
1  #include<stdio.h>
2  #include<conio.h>
3  #include<string.h>
4
5  int main()
6  {
7      int count=0,count1=0,count2=0,k=0,g=0;
8      int a,b,i,j,c,x;
9      char *ptr;
10     char *ptr1,*ptr2,*ptr3;
11
12     char temp[50],sim1[50],temp1[50];
13     char name[50],fname[50],sim[50];
14     char str[100],str1[100];
15
16     printf("Enter your name : ");
17     scanf("%s",&name);
18     printf("Enter your friend's name : ");
19     scanf("%s",&fname);
20     ptr=&name;
21     ptr1=&fname;
22
23     a=strlen(ptr);
24     b=strlen(ptr1);
25
26     for(i=0;name[i]!='\0';++i)
27     {
28         for(j=0;fname[j]!='\0';++j)
29         {
30             if(name[i]==fname[j])
31                 sim[k++]=name[i];
32             ptr1++;
33         }
34         ptr++;
35     }
36 }
```

```
37     c=k;
38     printf("\n");
39     for (int i = 0; i < c; i++)
40     {
41         for (j = 0; j < count2; j++)
42         {
43             if (sim[i] == temp[j])
44                 break;
45         }
46         if (j == count2)
47         {
48             temp[count2] = sim[i];
49             count2++;
50         }
51     }
52
53     printf("Similar characters: ");
54     for (int i = 0; i < count2; i++)
55         printf("%c ", temp[i]);
56     printf("\n");
57     printf("Similar characters count : %d",count2);
58     printf("\n");
59     strcpy(str,name);
60     strcpy(str1,fname);
61     strcat(str,str1);
62     x=strlen(str);
63     ptr2=&str;
64     ptr3=&sim;
65
66     for(i=0;str[i]!='\0';++i)
67     {
68         for(j=0;sim[j]!='\0';++j)
69         {
70             if(str[i]!=sim[j])
71                 sim1[g++]=str[i];
72             ptr3++;
```



```

73     }
74     ptr2++;
75 }
76
77 printf("\n");
78 for (int i = 0; i < g; i++)
79 {
80     for (j = 0; j < count1; j++)
81     {
82         if (sim1[i] == temp1[j])
83             break;
84     }
85     if (j == count1)
86     {
87         temp1[count1] = sim1[i];
88         count1++;
89     }
90 }
91
92 printf("Dissimilar characters count :%d",count1-count2);
93 printf("\n");
94 return 0;
95 }
96

```

### Output

```

Enter your name : Prashanth
Enter your friend's name : Praveen

Similar characters: P r a n
Similar characters count : 4

Dissimilar characters count :5

...Program finished with exit code 0
Press ENTER to exit console.

```

5) Write a C program to get the temperature of m days for n cities using dynamic 2D array. Calculate the maximum and minimum temperature of each day.

**Code :**

```
1  #include<stdio.h>
2  #include<conio.h>
3  #include<stdlib.h>
4  #include<limits.h>
5
6  int main() {
7      // m days(rows) and n cities(columns)
8      int m,n;
9      printf("Enter the number of days : ");
10     scanf("%d",&m);
11
12     printf("Enter the number of cities : ");
13     scanf("%d",&n);
14
15     int *temp = (int *)malloc(m*n*sizeof(int));
16
17     for(int i=0;i<m;i++) {
18         printf("Enter the temperature(celsius) of Day-%d : ",i+1);
19         for(int j=0;j<n;j++) {
20             scanf("%d",&*(temp + i*n + j));
21         }
22     }
23
24     int min = INT_MIN;
25     int max = INT_MAX;
26
27     for(int i=0;i<m;i++) {
28         for(int j=0;j<n;j++) {
29             if (*(temp + i*n + j)<max) { max = *(temp + i*n + j); }
30             else if (*(temp + i*n + j)>min) { min = *(temp + i*n + j); }
31         }
32         printf("\nDay-%d Maximum temperature(celsius) : %d",i+1,min);
33         printf("\nDay-%d Minimum temperature(celsius) : %d",i+1,max);
34         printf("\n\n");
35
36         min = INT_MIN;
37         max = INT_MAX;
38     }
39     return 0;
40 }
```

### Output:

```
Enter the number of days : 4
Enter the number of cities : 3
Enter the temperature(celsius) of Day-1 : 30 32 29
Enter the temperature(celsius) of Day-2 : 33 31 35
Enter the temperature(celsius) of Day-3 : 32 36 30
Enter the temperature(celsius) of Day-4 : 28 35 29

Day-1 Maximum temperature(celsius) : 32
Day-1 Minimum temperature(celsius) : 29

Day-2 Maximum temperature(celsius) : 35
Day-2 Minimum temperature(celsius) : 31

Day-3 Maximum temperature(celsius) : 36
Day-3 Minimum temperature(celsius) : 30

Day-4 Maximum temperature(celsius) : 35
Day-4 Minimum temperature(celsius) : 28

Process returned 0 (0x0) execution time : 66.711 s
Press any key to continue.
```

6) Tell me about how efficient your program is. (1 mark to show up your individuality, answer should be sound and sensible – Answers repeated no marks, so don't share your answers)

### Code

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int A[3][3] = {90,91,92,80,81,82,70,71,72};
5
6  void Display(int n) {
7      for(int i=0;i<n;i++) {
8          for(int j=0;j<n;j++) {
9              printf("%d ",A[i][j]);
10         }
11         printf("\n");
12     }
13 }
14
15 void Display_pointers(int n) {
16     int *ptr;
17     ptr = &A[0][0]; // points to the 1st row of the array (1st 1D array)
18     // ptr = A[0];
19
20     for(int i=0;i<3;i++) {
21         printf("%d ",*ptr);
22         ptr++;
23     }
24 }
25
26 void Represent_pointers() {
27     int *ptr;
28     ptr = A[0];
29     printf("\nptr : %d",ptr);
30
31     printf("\nA : %d",A);
32     printf("\n*A : %d",*A);
33     printf("\n&A : %d",&A);
34
35     printf("\n&A[0][0] : %d",&A[0][0]);
36     printf("\nA[0] : %d",A[0]);
```

```

37 }
38
39 void Next() {
40     printf("\nA+1 : %d",A+1);
41     printf("\n*(A+1): %d",*(A+1));
42     printf("\n&A[1] : %d",&A[1]);
43
44     printf("\n&A[1][0] : %d",&A[1][0]);
45     printf("\nA[1] : %d",A[1]);
46 }
47
48 void Element(int i,int j) {
49     printf("\nUsing array indexing : %d",A[i][j]);
50     printf("\nUsing pointers : %d", * (*(A+i)+j) );
51 }
52
53 int main() {
54

```

```

55     printf("Normal display\n");
56     Display(3);
57
58     printf("\nDisplaying with pointers\n");
59     Display_pointers(3);
60
61     printf("\n\nDifferent methods of printing pointers");
62     Represent_pointers();
63
64     printf("\n\nPointing to the next in array");
65     Next();
66
67     printf("\n\nElement at a particular index using pointers");
68     Element(1,2);
69
70     /* Question ???
71     *(a+1) -->
72     *(*a+1) -->
73     */
74     printf("\n\n*(A+1) : %d",*(A+1));
75     printf("\n\n*(A+1) : %d",*(A+1));
76     printf("\n\n**A : %d",**A);
77
78     int *ptr;
79     ptr = &A[0][0];
80     //printf("\n**ptr : %d",**ptr); --> ptr --> Arr[0][0], *ptr prints the value --> Arr[0][0], **ptr --> error
81
82     /* Question ???
83     a[1]+1 -->
84     &a[1]+1 -->
85     */
86
87     printf("\n\n1D array");
88     int arr1[] = {45,46,47};
89     printf("\narr1 : %d",arr1);
90     printf("\n&arr1 : %d",&arr1);

```

```

91     printf("\narr1 + 1 : %d",arr1+1);
92     printf("\n&arr1 + 1 : %d",&arr1+1); // (complete array --> 12)
93
94     // back to the question
95     printf("\n\nA[1] + 1 : ,%d",A[1]+1);
96     printf("\n&A[1] + 1 : ,%d",&A[1]+1);
97     // &arr1[1] --> (Complete 2nd 1-D array --> 12) and then points to the 3rd 1-D array
98     printf("\n(&A[1]+1) : %d",(&A[1]+1));
99
100    return 0;
101 }
102

```

## Output

```

Normal display
90 91 92
80 81 82
70 71 72

Displaying with pointers
90 91 92

Different methods of printing pointers
ptr : 4210720
A : 4210720
*A : 4210720
&A : 4210720
&A[0][0] : 4210720
A[0] : 4210720

Pointing to the next in array
A+1 : 4210732
*(A+1): 4210732
&A[1] : 4210732
&A[1][0] : 4210732
A[1] : 4210732

Element at a particular index using pointers
Using array indexing : 82
Using pointers : 82

*(A+1) : 4210732
*(*A+1) : 91
**A : 90

1D array
arr1 : 6422028
&arr1 : 6422028
arr1 + 1 : 6422032
&arr1 + 1 : 6422040

A[1] + 1 : ,4210736
&A[1] + 1 : ,4210744
(&A[1]+1) : 4210744
Process returned 0 (0x0) execution time : 5.963 s
Press any key to continue.

```

I used these array of pointers concepts in the 2<sup>nd</sup> and 3<sup>rd</sup> program.  
And also created dynamic array using pointers in the 5<sup>th</sup> program.

Github Link → <https://github.com/PrashanthSingaravelan/WinterSemester-2021/tree/main/CSE2010%20Advanced%20C%20programming/Assignment-2>