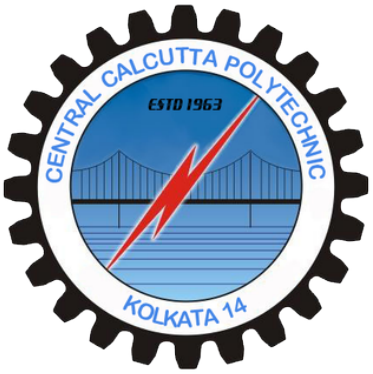

CENTRAL CALCUTTA POLYTECHNIC

21, Convent Road, Philips, Sealdah, Kolkata, West Bengal 700014

DEPT. : COMPUTER SCIENCE AND TECHNOLOGY

- NAME : ASIM BERA
- ROLL : DCCPCSTS3
- NO : 10005504
- SUBJECT : PROGRAMMING IN C
- SESSION : 2020 - 21

Contents

1 Single Dimensional Arrays

1.1 Write a C program to read and print elements of array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }
  // print elements
  printf("Elements :\n");
  for (int i = 0; i < b; i++)
  {
    printf("%d ", a[i]);
  }
  printf("\n");

  return 0;
}
```

Program Output :

```
➜ gcc 01.c && ./a.out
Enter number of elements => 5
Element 0 => 45
Element 1 => 63
Element 2 => 52
Element 3 => 45
Element 4 => 36
Elements :
45 63 52 45 36
```

1.2 Write a C program to print all negative elements in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }
  // print elements
  printf("Negative Elements :\n");
  for (int i = 0; i < b; i++)
  {
    if (a[i] < 0)
      printf("Element %d => %d\n", i, a[i]);
  }

  return 0;
}
```

Program Output :

```
➜ gcc 02.c && ./a.out
Enter number of elements => 6
Element 0 => 56
Element 1 => -23
Element 2 => -56
Element 3 => -78
Element 4 => 45
Element 5 => 65
Negative Elements :
Element 1 => -23
Element 2 => -56
Element 3 => -78
```

1.3 Write a C program to find sum of all array elements.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b, sum = 0;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }
  // print elements
  for (int i = 0; i < b; i++)
  {
    sum += a[i];
  }

  printf("Sum = %d\n", sum);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 03.c && ./a.out && flameshot gui
Enter number of elements => 6
Element 0 => 56
Element 1 => -41
Element 2 => 23
Element 3 => 45
Element 4 => 96
Element 5 => 78
Sum = 257
```

1.4 Write a C program to find maximum and minimum element in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b, max, min;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }

  for (int i = 0; i < b; i++)
  {
    if (i != 0)
    {
      if (max < a[i])
        max = a[i];
      if (min > a[i])
        min = a[i];
    }
    else
    {
      max = a[i];
      min = a[i];
    }
  }

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

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 04.c && ./a.out
Enter number of elements => 7
Element 0 => 23
Element 1 => -56
Element 2 => 78
Element 3 => 999
Element 4 => 120
Element 5 => -223
Element 6 => 52
Maximum: 999, Minimum: -223
```

1.5 Write a C program to count total number of even and odd elements in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b, even = 0, odd = 0;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }

  for (int i = 0; i < b; i++)
  {
    if (a[i] % 2 == 0)
      even += 1;
    else
      odd += 1;
  }

  printf("Even: %d, Odd: %d\n", even, odd);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 05.c && ./a.out
Enter number of elements => 6
Element 0 => 23
Element 1 => 71
Element 2 => -23
Element 3 => 46
Element 4 => 32
Element 5 => -22
Even: 3, Odd: 3
```

1.6 Write a C program to count total number of negative elements in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int b, n = 0;
  printf("Enter number of elements => ");
  scanf("%d", &b);

  // take input
  for (int i = 0; i < b; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }

  for (int i = 0; i < b; i++)
  {
    if (a[i] < 0)
      n += 1;
  }

  printf("No of Negetive Elements: %d \n", n);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 06.c && ./a.out
Enter number of elements => 8
Element 0 => 89
Element 1 => -65
Element 2 => 54
Element 3 => -28
Element 4 => -98
Element 5 => 12
Element 6 => -88
Element 7 => 65
No of Negetive Elements: 4
```

1.7 Write a C program to copy all elements from an array to another array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50], b[50];
  int c;
  printf("Enter number of elements (array a) => ");
  scanf("%d", &c);

  // take input
  for (int i = 0; i < c; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }

  for (int i = 0; i < c; i++)
  {
    b[i] = a[i];
  }
  printf("Elements of array b => ");
  for (int i = 0; i < c; i++)
  {
    printf("%d ", b[i]);
  }
  printf("\n");

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 07.c && ./a.out
Enter number of elements (array a) => 4
Element 0 => 23
Element 1 => 56
Element 2 => 74
Element 3 => 32
Elements of array b => 23 56 74 32
```

1.8 Write a C program to insert an element in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50];
  int l, d, e;
  printf("No of elements: ");
  scanf("%d", &l);

  for (int i = 0; i < l; i++)
  {
    printf("Element (%d): ", i);
    scanf("%d", &a[i]);
  }

  printf("Element to insert => ");
  scanf("%d", &d);
  printf("Element position=> ");
  scanf("%d", &e);

  if (e > (l - 1))
  {
    printf("Position is out of range.");
    return 1;
  }

  for (int i = l - 1; i >= e; i--)
  {
    a[i + 1] = a[i];
  }
  a[e] = d;

  printf("Resulting array => ");
  for (int i = 0; i < (l + 1); i++)
  {
    printf("%d ", a[i]);
  }
  printf("\n");

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04
➜ gcc 08.c && ./a.out
No of elements: 6
Element (0): 45
Element (1): 32
Element (2): 85
Element (3): 63
Element (4): 71
Element (5): 38
Element to insert => 999
Element position=> 3
Resulting array => 45 32 85 999 63 71 38
```

1.9 Write a C program to delete an element from an array at specified position.

Source Code :

```c
// Delete from specified position
#include <stdio.h>

int main()
{
  int a[50];
  int l, e;
  printf("No of elements: ");
  scanf("%d", &l);

  for (int i = 0; i < l; i++)
  {
    printf("Element (%d): ", i);
    scanf("%d", &a[i]);
  }

  printf("Element position=> ");
  scanf("%d", &e);

  if (e > (l - 1))
  {
    printf("Position is out of range.");
    return 1;
  }

  // shift position
  for (int i = e; i < l; i++)
  {
    a[i] = a[i + 1];
  }
  // reduce array length by 1
  l--;

  printf("Resulting array => ");
  for (int i = 0; i < l; i++)
  {
    printf("%d ", a[i]);
  }
  printf("\n");

  return 0;
}
```

Program Output :

```
➜ gcc 09.c && ./a.out
No of elements: 5
Element (0): 65
Element (1): 23
Element (2): 78
Element (3): -56
Element (4): 12
Element position=> 1
Resulting array => 65 78 -56 12
```

1.10 Write a C program to merge two array to third array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int arr_a[50], arr_b[50], arr_c[100], l, p;
  float avg;

  printf("No of elements for arr_a: ");
  scanf("%d", &l);
  for (int i = 0; i < l; i++)
  {
    printf("Elem (%d): ", i);
    scanf("%d", &arr_a[i]);
  }

  printf("No of elements for arr_b: ");
  scanf("%d", &p);
  for (int i = 0; i < p; i++)
  {
    printf("Elem (%d): ", i);
    scanf("%d", &arr_b[i]);
  }

  int q = l + p;
  for (int i = 0; i < q; i++)
  {
    if (i < l)
      arr_c[i] = arr_a[i];
    else
      arr_c[i] = arr_b[i - l];
  }

  printf("Elems of arr_c: \n");
  for (int i = 0; i < q; i++)
  {
    printf("%d ", arr_c[i]);
  }

  return 0;
}
```

Program Output :

```
➜ gcc 10.c && ./a.out
No of elements for arr_a: 3
Elem (0): 45
Elem (1): 69
Elem (2): 78
No of elements for arr_b: 2
Elem (0): 56
Elem (1): 84
Elems of arr_c:
45 69 78 56 84
```

1.11 Write a C program to find reverse of an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int a[50], b[50];
  int l;
  printf("Enter number of elements => ");
  scanf("%d", &l);

  // take input
  for (int i = 0; i < l; i++)
  {
    printf("Element %d => ", i);
    scanf("%d", &a[i]);
  }

  // reverse the array
  for (int i = l - 1; i >= 0; i--)
  {
    b[l - i - 1] = a[i];
  }

  // print elements
  printf("Elements :\n");
  for (int i = 0; i < l; i++)
  {
    printf("%d ", b[i]);
  }

  return 0;
}
```

Program Output :

```
➜ gcc 11.c && ./a.out
Enter number of elements => 4
Element 0 => 36
Element 1 => 85
Element 2 => 3
Element 3 => 41
Elements :
41 3 85 36
```

1.12 Write a C program to put even and odd elements of array in two separate array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int arr_a[100],
      // array of even numbers
      arr_b[50],
      // array of odd numbers
      arr_c[50],
      // l = first array length, p = pointer of arr_b, q = pointer of arr_c
      l, p = 0, q = 0;

  printf("No of elements: ");
  scanf("%d", &l);
  for (int i = 0; i < l; i++)
  {
    printf("Elem (%d): ", i);
    scanf("%d", &arr_a[i]);
  }

  for (int i = 0; i < l; i++)
  {
    if (arr_a[i] % 2 == 0)
    {
      arr_b[p] = arr_a[i];
      p++;
    }
    else
    {
      arr_c[q] = arr_a[i];
      q++;
    }
  }

  printf("\nEven array: \n");
  for (int i = 0; i < p; i++)
  {
    printf("%d ", arr_b[i]);
  }
  printf("\nOdd array: \n");
  for (int i = 0; i < q; i++)
  {
    printf("%d ", arr_c[i]);
  }

  return 0;
}
```

Program Output :

```
➜ gcc 12.c && ./a.out
No of elements: 5
Elem (0): 32
Elem (1): 45
Elem (2): 66
Elem (3): 20
Elem (4): 12

Even array:
32 66 20 12
Odd array:
45
```

1.13 Write a C program to search an element in an array.

Source Code :

```c
#include <stdio.h>

int main()
{
  int arr[50], l, q;

  printf("No of elements: ");
  scanf("%d", &l);
  for (int i = 0; i < l; i++)
  {
    printf("Elem (%d): ", i);
    scanf("%d", &arr[i]);
  }

  printf("Element to search: ");
  scanf("%d", &q);

  for (int i = 0; i < l; i++)
  {
    if (arr[i] == q)
    {
      printf("Element Found. Index: %d\n", i);
      return 0;
    }
  }

  printf("Element not found.\n");

  return 0;
}
```

Program Output :

```
➜ gcc 13.c && ./a.out
No of elements: 5
Elem (0): 85
Elem (1): 36
Elem (2): 45
Elem (3): 6
Elem (4): 21
Element to search: 45
Element Found. Index: 2
```

1.14 Write a C program to sort array elements in ascending or descending order.

Source Code :

```c
#include <stdio.h>
#include <limits.h>

int main()
{
  int a[50];
  int l, order;
  printf("No of elements: ");
  scanf("%d", &l);

  // take input
  for (int i = 0; i < l; i++)
  {
    printf("Element (%d): ", i);
    scanf("%d", &a[i]);
  }

  printf("\nOrder (0 = ascending, 1 = descending): ");
  scanf("%d", &order);

  // selectionsort
  for (int i = 0; i < l; i++)
  {
    int m = a[i], n = !order ? INT_MAX : INT_MIN, o = i;
    for (int j = i; j < l; j++)
    {
      if (!order ? n > a[j] : n < a[j])
      {
        n = a[j];
        o = j;
      }
    }
    a[i] = n;
    a[o] = m;
  }

  // print all elements
  printf("\nElements :\n");
  for (int i = 0; i < l; i++)
  {
    printf("%d ", a[i]);
  }

  return 0;
}
```

Program Output :

```
➜ gcc 14.c && ./a.out
No of elements: 5
Element (0): 23
Element (1): 45
Element (2): 68
Element (3): -98
Element (4): 21

Order (0 = ascending, 1 = descending): 1

Elements :
68 45 23 21 -98
```