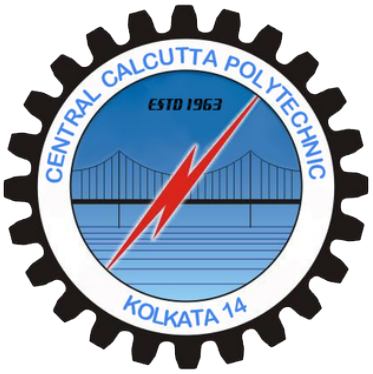

CENTRAL CALCUTTA POLYTECHNIC

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
  printf("Enter array Elements: ");
  for (int i = 0; i < b; 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 :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 6s
➜ gcc 01.c && ./a.out
Enter number of elements: 6
Enter array Elements: 11 52 65 85 45 65
Elements :
11 52 65 85 45 65
```

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
  printf("Enter array Elements: ");
  for (int i = 0; i < b; i++)
    scanf("%d", &a[i]);
  // print elements
  printf("Negative Elements: ");
  for (int i = 0; i < b; i++)
  {
    if (a[i] < 0)
      printf("%d ", a[i]);
  }

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 12s
➜ gcc 02.c && ./a.out
Enter number of elements => 7
Enter array Elements: 32 -85 45 52 -12 -78 520
Negative Elements: -85 -12 -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
  printf("Enter array Elements: ");
  for (int i = 0; i < b; 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 on  main [!?] took 28s
➜ gcc 03.c && ./a.out
Enter number of elements => 5
Enter array Elements: 10 56 32 45 -32
Sum = 111
```

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

Source Code :

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

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

  // take input
  printf("Enter array Elements: ");
  for (int i = 0; i < b; i++)
    scanf("%d", &a[i]);

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

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

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 13s
➜gcc 04.c && ./a.out
Enter number of elements => 6
Enter array Elements: 45 -23 56 85 456 -215
Maximum: 456, Minimum: -215
```

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
  printf("Enter array Elements: ");
  for (int i = 0; i < b; 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 on  main [!?] took 25s
➜ gcc 05.c && ./a.out
Enter number of elements => 5
Enter array Elements: 45 22 65 35 74
Even: 2, 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
  printf("Enter array Elements: ");
  for (int i = 0; i < b; i++)
    scanf("%d", &a[i]);

  for (int i = 0; i < b; i++)
    n += a[i] < 0 ? 1 : 0;

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

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 21s
➜gcc 06.c && ./a.out
Enter number of elements => 6
Enter array Elements: 12 -25 36 -85 -65 16
No of Negetive Elements: 3
```

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
  printf("Enter array Elements: ");
  for (int i = 0; i < c; 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]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 50s
➜ gcc 07.c && ./a.out
Enter number of elements (array a) => 8
Enter array Elements: 12 56 45 85 35 25 74 36
Elements of array b => 12 56 45 85 35 25 74 36
```

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);

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

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

  if (e > (l - 1) || e < 0)
  {
    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]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 23s
➜ gcc 08.c && ./a.out
No of elements: 6
Enter array Elements: 14 56 85 325 45 41
Element to insert => -21
Element position=> 2
Resulting array => 14 56 -21 85 325 45 41
```

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);

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

  printf("Element position to delete: ");
  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]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?]
➜gcc 09.c && ./a.out
No of elements: 5
Enter array Elements: 45 25 36 85 14
Element position to delete: 3
Resulting array => 45 25 36 14
```

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);

  printf("Elements for arr_a: ");
  for (int i = 0; i < l; i++)
    scanf("%d", &arr_a[i]);

  printf("No of elements for arr_b: ");
  scanf("%d", &p);
  printf("Elements for arr_b: ");
  for (int i = 0; i < p; 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 :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 17s
➜ gcc 10.c && ./a.out
No of elements for arr_a: 3
Elements for arr_a: 25 32 -56
No of elements for arr_b: 4
Elements for arr_b: 56 25 45 36
Elems of arr_c:
25 32 -56 56 25 45 36
```

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
  printf("Enter array elements: ");
  for (int i = 0; i < l; 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("Reversed Array: ");
  for (int i = 0; i < l; i++)
    printf("%d ", b[i]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 8s
➜gcc 11.c && ./a.out
Enter number of elements: 6
Enter array elements: 89 52 1 5 89 65
Reversed Array: 65 89 5 1 52 89
```

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[100], arr_even[50], arr_odd[50], arr_len, p = 0, q = 0;

  printf("No of elements: ");
  scanf("%d", &arr_len);
  printf("Enter array elements: ");
  for (int i = 0; i < arr_len; i++)
    scanf("%d", &arr[i]);

  for (int i = 0; i < arr_len; i++)
  {
    if (arr[i] % 2 == 0)
      arr_even[p++] = arr[i];
    else
      arr_odd[q++] = arr[i];
  }

  printf("\nEven array: ");
  for (int i = 0; i < p; i++)
    printf("%d ", arr_even[i]);

  printf("\nOdd array: ");
  for (int i = 0; i < q; i++)
    printf("%d ", arr_odd[i]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 15s
➜ gcc 12.c && ./a.out
No of elements: 6
Enter array elements: 25 28 32 45 65 92

Even array: 28 32 92
Odd array: 25 45 65
```

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);
  printf("Enter array elements: ");
  for (int i = 0; i < l; 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 :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 16s
➜ gcc 13.c && ./a.out
No of elements: 6
Enter array elements: 45 25 36 85 78 69
Element to search: 85
Element Found. Index: 3
```

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
  printf("Enter array elements: ");
  for (int i = 0; i < l; i++)
    scanf("%d", &a[i]);

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

  // selection sort
  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;
  }

  printf("Sorted Array: ");
  for (int i = 0; i < l; i++)
    printf("%d ", a[i]);

  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 47s
➜ gcc 14.c && ./a.out
No of elements: 5
Enter array elements: 23 12 -56 85 0
Sort Order (0 = ascending, 1 = descending): 0
Sorted Array: -56 0 12 23 85
```

1.15 Write a C program to left rotate an array by n positions.

Source Code :

```c
#include <stdio.h>

int main()
{
  int arr[100], n, r, temp;
  printf("Enter no elements: ");
  scanf("%d", &n);
  printf("Enter array elements: ");
  for (int i = 0; i < n; i++)
    scanf("%d", &arr[i]);
  printf("Enter rotation times: ");
  scanf("%d", &r);
  for (int i = 0; i < r; i++)
  {
    temp = arr[0];
    for (int j = 0; j < n; j++)
      arr[j] = arr[j + 1];
    arr[n - 1] = temp;
  }
  printf("Result: ");
  for (int j = 0; j < n; j++)
    printf("%d ", arr[j]);
  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 6s
➜ gcc 15.c && ./a.out
Enter no elements: 10
Enter array elements: 12 56 85 45 69 78 120 45 96 70
Enter rotation times: 4
Result: 69 78 120 45 96 70 12 56 85 45
```

1.16 Write a C program to right rotate an array by n postions.

Source Code :

```c
#include <stdio.h>

int main()
{
  int arr[100], n, r, temp;
  printf("Enter no elements: ");
  scanf("%d", &n);
  printf("Enter array elements: ");
  for (int i = 0; i < n; i++)
    scanf("%d", &arr[i]);
  printf("Enter rotation times: ");
  scanf("%d", &r);
  for (int i = 0; i < r; i++)
  {
    temp = arr[n - 1];
    for (int j = n - 1; j >= 0; j--)
      arr[j] = arr[j - 1];
    arr[0] = temp;
  }
  printf("Result: ");
  for (int j = 0; j < n; j++)
    printf("%d ", arr[j]);
  return 0;
}
```

Program Output :

```
ccp-assignments/c_lang/assignment_04 on  main [!?] took 5s
➜ gcc 16.c && ./a.out
Enter no elements: 5
Enter array elements: 1 2 3 4 5
Enter rotation times: 1
Result: 5 1 2 3 4
```