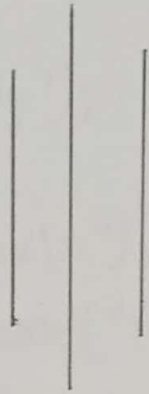


# KATHMANDU UNIVERSITY

Department of Computer Engineering.



A

Lab Report On

Computer Programming (COMP102)

Lab Sheet No: 4

Submitted by:

Ashraya Kadel

UNQ CE I/I

Roll No: 25

Submitted to:

Sameer Tamrakar

Department of

Computer Engineering

Submission date: 12/04/2023

<Q.17>: WAP to check odd numbers in the array.

Ans:

\*) Source Code:

```
#include <stdio.h>
void main()
{
    int A[15], i, n, j;
    printf("Numbers to apply");
    scanf("%d", &n);
    printf("Enter the numbers\n");
    for (i=0; i<n; i++) scanf("%d", &A[i]);
    printf("Odd numbers:");
    for (j=0; j<n; j++)
        if (A[j] % 2 != 0) printf("%d\n", A[j]);
}
```

\*) Output

```
Numbers to apply
5
Enter the numbers
1
98
6
7
8
Entered odd numbers
1
7
```

\*) Description

This program reads numbers in an array and checks for the smallest number.

Q.2: WAP to reverse elements of the array.  
Ans:

\* Source Code

```
#include <stdio.h>
void main()
{
    int A[10], i, j, n, k;
    printf("Enter size of array\n");
    scanf("%d\n", &n);
    printf("Enter the numbers\n");
    for (i=0; i<n; i++) scanf("%d", &A[i]);
    printf("Reverse of array:\n");
    k=n-1;
    for (j=k; j>=0; j--) printf("%d\n", A[j]);
}
```

\* Output

```
Enter size of array
4
Enter the numbers
1
3
5
7
Reverse of array
7
5
3
1
```

\* Description

This program reads the elements of the array and display it in reverse order.

<Q.3>: WAP to display unique elements in an array.

\* Source code:

```
#include <stdio.h>
void main()
{
    int A[10], i, j, k, n, m, count = 0;
    printf("Enter array size\n");
    scanf("%d", &n);
    printf("Enter the numbers\n");
    for (i = 0; i <= (n-1); i++) scanf("%d", &A[i]);
    printf("The unique element\n");
    for (j = 0; j <= (n-1); j++)
    {
        count = 0;
        for (m = 0; m <= (n-1); m++)
        {
            if (A[j] == A[m]) count = count + 1;
        }
        if (count == 1) printf("%d\n", A[j]);
    }
}
```

\* Output:

```
Enter array size
6
Enter the numbers
1
1
2
3
4
4
The unique element
2
3
```

\* Description

This program reads the numbers in the array and reads for the elements that are not repeated. Here, 2 and 3 is displayed as output.



Q.47: WAP to find smallest element of an array.

Ans

\* Source code

```
#include <stdio.h>
void main()
{ int A[10], i, n, j, small;
  printf("Number of array | n");
  scanf("%d", &n);
  printf("Enter the numbers \n");
  for (i=0; i<=(n-1); i++) scanf("%d", &A[i]);
  small = A[0];
  for (j=1; j<=(n-1); j++)
  { if (small > A[j]) small = A[j]; }
  printf("The smallest number = %d", small);
}
```

\* Output

Number of array

4

Enter the number

32

33

34

35

The smallest number = 32

\* Description

This program reads numbers in an array and it displays the smallest number

Q.5: WAP to find LCM of two elements.

Ans:

\*) Source Code

```
#include <stdio.h>
int lcm (int n1, int n2);
void main ()
{ int n1, n2;
  lcm (n1, n2);
}
int lcm (int n1, int n2)
{ int i, c=1, b;
  printf("Enter numbers\n"); scanf("%d", &n1, &n2);
  for (i=1, i<=n1; i++)
  { if (n1%i==0 & n2%i==0) c=i; }
  b = (n1*n2)/c;
  printf("LCM is %d", b);
}
```

\*) Output

Enter numbers  
2  
6  
LCM is 6.

\*) Description

This program calls for a function to calculate the least common multiple.

Q.6: WAP to sort array with recursion:

Ans:

\* Source Code

```
#include <stdio.h>
void sort(int A[], int);
void main()
{ int A[10], i, n;
  printf("Enter size\n"); scanf("%d", &n);
  printf("Enter the elements\n");
  for (i=0; i<n; i++) scanf("%d", &A[i]);
  sort(A, n);
  printf("Sorted array");
  for (i=0; i<n; i++) printf("%d\t", A[i]); }

void sort(int A[], int n)
{ int i, d;
  if (n==1) return;
  for (i=0; i<n; i++)
  { if (A[i] > A[i+1])
    { d = A[i];
      A[i] = A[i+1];
      A[i+1] = d;
    } }
  n--;
  sort(A, n); }
```

\* Description

This program using recursion to sort out an array in ascending order.

\* Output

```
Enter size    4
Enter the elements
90  10  70  20
Sorted array
10  20  70  90
```

Q7: Display pattern:

Ans:



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

```
void main()
```

```
{ int i, j, k;
```

```
for (i=1; i<=5; i++)
```

```
{ for (k=1; k<=5-i; k++)
```

```
{ printf(" ");
```

```
}
```

```
for (j=1; j<=i; j++)
```

```
{ printf(" *");
```

```
printf("\n");
```

```
}
```

```
}
```



Q.8: WAP to check palindrome.

Ans:

\* Source Code:

```
#include <stdio.h>
#include <string.h>
void main()
{ char A[10];
  int i, n, k=0;
  gets(A);
  n=strlen(A);
  for (i=0; i<n/2; i++)
  {
    if (A[i] == A[n-i-1])
      k++;
  }
  if (k==i) printf("%s is palindrome", A);
  else printf("%s is not palindrome", A);
}
```

\* Output

Level

Level is palindrome.

\* Description

This program checks from the string palindrome string.