CIVIL

Name : Hathila Vivekkumar Vrameshbhai

Roll no = 25BCL025

M T W T F S S

Page No.:

Date:

YOUVA

Assignment - 3

Section - A

→ Programs using array:

(1) Accept 5 values and print them later on

(A)

```
#include <stdio.h>
int main()
{
    int i, a[5];
    printf("Enter 5 value:");
    for (i=0; i<5; i++)
    {
    scanf ("%d", &a[i]);
    }
    printf ("\n the 5 values are:");
    for (i=0, i<5; i++)
    {
    printf ("%d", a[i]);
    }
    return 0;
    }
```

⇒ Output : Enter 5 values : 23, 45, 24, 32, 14

The 5 value are : 23, 45, 24, 14

(2) Accept 10 value and print 4th, 7th and 9th velue

(A)
```
# include <stdio.h>
int main ()
{
    int i, a[10];
    printf ("Enter 10 values: ");
    for (i=0; i<10; i++)
    {
    scanf ("%d", &a[i]);
    }
printf ("4th velue = %d\n", a[3]);
printf ("7th value = %d\n", a[6]);
printf ("9th value = %d\n", a[8]);
return 0;
    }
```

→ out put:

Enter 10 values: 43, 52, 46, 72, 76
86, 45, 61, 35, 65

4th value = 72
7th value = 45
9th value = 35

(3) Accept is values and sort the corrau in ascending / descendince order find out different techniques for sorting and intify the best ones

(A)

```
#include <stdio.h>
int main()
{
    int a[5], i, j, temp;
    printf("Enter 5 value:");
    for (i=0; i<5; i++)
    {
        scanf("%d", &a[i]);
    }
    for (i=0; i<5; i++)
    {
        for (j=i+1; i<5; j++)
        {
            if (a[i] > a[j])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
    printf("Ascending order:");
    for (i=0; i<5; i++)
    { printf("%d", a[i]); }
    printf("\n Descending order:");
    for (i=4; i>=0; i--)
    { printf("%d", a[i]); }
    return 0;
}
```

=) output :-

Enter 5 value : 33, 56, 98, 67, 13
Ascending order : 13, 33, 56, 67, 98
Descending order : 98, 67, 56, 33, 13

(4) Print minimum number of notes required

(A)
```
#include<stdio.h>
int main()
{
    int amount, i, count[9] = {0};
int notes[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};

printf("Enter the amount:");
scanf("%d", &amount);
for(i=0; i<9; i++)
{
count[i] = amount / notes[i];
amount = amount % notes[i];
}
printf("Minimum notes required:\n");
for(i=0; i<9; i++)
{
printf("%d * %d \n", count[i], notes[i]);
}
return 0;
```

=) Output: Enter the amount: 28

Minimum notes required:

1 * 20

1 * 5

1 * 2

1 * 1

(5) Add two arrays of same size and store the result in the 3rd array

(A)

```
#include<stdio.h>
int main()
{
int i,j, a[2][2], b[2][2], sum[2][2];
printf("Enter first 4 value: \n");
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{
scanf("%d", &a[i][j]);
}
printf("Enter second 4 value: \n");
for(i=0; i<2; i++)
{
for(j=0; j<2; j++)
{ scanf("%d", &b[i][j]); }
}
for(i=0; i<2; i++)
```

```
    for (j=0; j<2; j++)
    {
    sum[i][j] = a[i][j] + b[i][j];
    }
  }
  printf ("%d", sum[i][j]);
  }
  printf ("\n");
  }
  return 0;
  }
```

⇒ output:

Enter first value: 2 3 4 5
Enter second value: 2 6 5 4
Sum of first and second:

(6) Multiply two 2D Arrays and store the result in the third one.

(A)

```
#include <stdio.h>
int main()
{
int i, j, k, a[2][2], b[2][2], m[2][2];
printf("Enter first values:\n");
for (i=0; i<2; i++)
```

```
{
  scanf("%d", &a[i][j]);
  }
}
printf("Enter second 4value:\n");
for(i=0; j<2; i++)
 {
 for(j=0; j<2; j++)
 {
 scanf("%d", &b[i][j]);
 }
}
   for(k=0, k<2; k++)
  {
  }
  printf("Multiplication of first and
  second :\n");
   for(j=0; j<2; j++)
}
printf("%d", mul[i][j]);
}
printf("\n");
}
return 0;
}
```

⇒ Output: Enter first 4values: 4 5 3 4
Enter second 4values: 3 6 2 5
Multiplication of first and second:
27 49
41 36

```
{
    scanf("%d", &a[i][j]);
}
printf("Enter second 4 value:\n");
for(i=0; j<2; i++)
{
for(j=0; j<2; j++)
{
scanf("%d", &b[i][j]);
}
}
    for(k=0, k<2; k++)
{
}
printf("Multiplication of first and
second :\n");
for(j=0; j<2; j++)
{
printf("%d", mul[i][j]);
}
printf("\n");
}
return 0;
}
```

⇒ Output: Enter first 4 values: 4 5 3 4
Enter second 4 values: 3 6 2 5
Multiplication of first and second:
52 49
41 36

(x) Obtain transpose of a 4×4 matrix.

(A)
```
#include <stdio.h>
int main()
{
  int i, j, matrix[4][4], transpose[4][4];
printf("Enter 16 value:\n");
 for(i=0; i<4; i++)
  {
  for(j=0 , j<4, j++)
  {
  scanf("%d", &matrix[i][j]);
  }
}
for(i=0; i<4; i++)
{
  transpose[j][i] = matrix[i][j];}
printf("Transpose of the matrix:\n");
for(i=0; i<4; i++)
{ for(j=0, j<4, j++)
printf("%d", transpose[i][j]);}
  printf("\n");
  return 0;
  }
```

Output:

Enter 16 value:
4 7 5 8 9 6 7 4 6 7 8 9 7 7 9 7

Transpose of the matrix:

4 9 6 7
7 6 7 9
5 7 7 3

(8) Copy one array of 5 elements to another array of elements

(A)

```
#include <stdio.h>
int main()
{
    int i, source[5], all[10];
    printf("Enter 5 elements of array:\n");
    for(i=0; i<5; i++)
    {
        scanf("%d", &source[i]);
    }
    for(i=0; i<10; i++)
    {
        all[i]=0;
    }
    for(i=0; i<10; i++);
    {
        printf("\n");
        return 0;
    }
```

=> Output: Enter 5 elements of array:
4 3 5 2 7
After copying of 5 Elements:
0 43 52 70 0 0 0

(9) Reverse an array of maximum 5 elements

```
#include <stdio.h>
int main()
{
int arr[5], i;
printf("Enter 5 elements:\n");
for(i=0; i<5; i++)
{
scanf("%d", &arr[i]);
}
printf("Reversed array:\n");
for(i=4; i>=0; i--)
{
printf("%d\n", arr[i]);
}
return 0;
}
```

→ Output: Enter 5 elements: 3 5 8 5 4
Reversed array: 4 5 8 5 3

(10) Find frequency of each number in an array.

(A)
```
#include <stdio.h>
int main()
{
int i, j, arr[10], freq[10];
printf("Enter 10 elements:\n");
```

```
for (i=0; i<10; i++)
{
  for (i=0, i<10; i++)
{
  for (j=i+1; j<10, j++)
{ if (arr[i] == arr[j])
{  count++;
  freq[j]=0;
}
}
 if (freq[i] != 0)
}
}
printf ("frequency of each elements
        \n");
}
for (i=0, i<10, i++)
{  if (freq[i] != 0)
{
 printf ("%d occurs %d times\n
  arr[i], freq[i]);
}
}
return 0;
}
```

⇒

Output :
Enter 10 elements :
3 5 4 6 5 4 6 2 8 7
Frequency of each elements

3 occurs 1 times
5 occurs 2 times
4 occurs 2 times
6 occurs 2 times
2 occurs 1 times
8 occurs 2 times
7 occurs 1 times

(11) shift all numbers by n positions within an array of 10 elements, pad with 0.

(A)
```
#include <stdio.h>
int main()
{
    int n, i, choice, temp[10] = {0},
    arr[10];

    printf("Enter 10 elements:\n");
    for(i=0; i<10; i++)
    {
        scanf("shift direction(1 for
        right) : ");
        scanf("%d", &choice);

        if(choise == 1)
        {
            for (i=0, i<10-n; i++)
        {
            temp[i] = arr[i+n];
        }
    }
```

```
else if (choice==2)
{
  for (i=n; i<10; i++)
  {
    temp[i]=arr[i-n];
  }
}
printf("shifted array = \n");
for (i=0; i<10; i++)
  printf("\n");
return 0;
}
```

=> output:

Enter 10 elements:
3 5 7 4 5 3 8 9 7 6
Enter number of positions to shift:
shift direction (1 for left 2 for right):
shift array: 0 0 0 0 0 0 3 5 7

(12) Insert a new number at the beginning of the array

(A)

```
#include <stdio.h>
int main()
{
  int i, n, newNum, arr[10];
  printf("Enter %d elements: \n", n);
  for (i=0; i<n; i++)
  {
```

```
for (i=n; i>0; i--)
{
arr[i] = arr[i-1];
}
arr[0] = new Num;
arr[0] = new Num;
n++;
{
printf("%d", arr[i]);
}
printf("\n");
return 0;
}
```

⇒ Output:

Enter of elements (max 10): 4
enter 4 elements: 3 5 27
Enter new number at first place: 3
Array after insertion: 3 3 5 27.

(13) Insert a new number at a particular position in array.

```
#include <stdio.h>
int main()
{
int i, n, pos, a[20];
printf("Enter number of elements:");
scanf("%d", &n);
```

printf ("Enter elements:");

{

printf ("Enter new number:");

scanf ("%d", &num);

for (i=n; i>=pos; i--)

{

a[i] = a[i-1];

}

a[pos-1] = num;

n++;

printf ("New array:");

for (i=0; i<n; i++)

{

printf ("%d", a[i]);

return 0;

}

=> Output: Enter number of elements

Enter elements: 5 8 6 3

Enter position: 2

Enter new number: 9

New array: 5 9 8 6 3

(iv) Insert a new number at the last position of a array.

(A) #include<stdio.h>

int main()

```
{
    int i, n, num, a[20];
    printf("Enter the number of elements");
    scanf("%d", &n);
    }
    printf("Enter new number to insert at last:");
    scanf("%d", &num);

    a[n] = num;
    n++;
    printf("New array:");
    for(i=0; i<n; i++)
    {printf("%d", a[i]);}
    return 0;
    }
```

output: Enter number of elements: 5
Enter elements: 3 6 4 8 9
Enter number to insert at last: 5
New array: 3 6 4 8 9 5

Delete value from the first position of an array

```
#include<stdio.h>
int main()
{
    int i, n, a[20];
    printf("Enter the number of elements");
    scanf("%d", &n);
```

```
printf("Enter elements:");
for(i=0;i<n;i++);
{ scanf("%d",&a[i]);}

for(i=0,i<n-1;i++)
{ a[i]=a[i+1];}

{ printf("%d",a[i]);}

return 0;
}
```

⇒ Output:- Enter number of elements = 4
Enter elements = 2 7 5 7
Array after deleting first element
7 5 7

(16) Delete a value from a particular position in a array

(A)
```
#include<stdio.h>
int main()
{
int i, n, pos, a[20];

printf("Enter number of elements");
scanf("%d",&n);

printf("Enter elements");
for(i=0;i<n;i++)
{
```

printf("New array\n");
for(i=0; i<n; i++)
{ printf("%d", a[i]); }
return 0;
}

=) output: Enter number of elements = 4
enter elements: 6 4 5 2
Enter position to delete = 3
the array: 6 4 7

(7) Delete as 'save' from the last position of an array

(A) #include<stdio.h>
int main()
{
int i, n, a[20];
printf("Enter number of elements: ");
scanf("%d", &n);

printf("Enter elements: ");
for(i=0; i<n; i++)
{ printf("%d", a[i]); }
return 0;
}

=) output: Enter number of elements.

Enter elements: 6 4 8 2 6

Array After deleting 1st elements
: 6 4 8 2

(18) Delete a value from the arrays.

(A)

```
#include <stdio.h>
int main()
{
    int i, j, n, num, a[20];
    printf("Enter number of elements");
    scanf("%d", &n);
    printf("Enter elements:");
    for(i=0; i<n; i++)
    { scanf("%d", &a[i]); }
    printf("New array:");
    for(i=0; i<n; i++)
    printf("%d", a[i]);
    return 0;
}
```

=) ~~order~~ Output

Enter number of elements : 4
Enter elements : 6 4 9 7
Enter value to delete : 9
New array : 6 4 7

(19) Search a value within an array

(11)
```
#include<stdio.h>
int main()
{
int i, n, num, a[20], found=0;
printf("Enter number of elements:");
scanf("%d", &n);
printf("Enter elements:");
for(i=0; i<n; i++)
{
if(a[i]==num)
{
found=1;
break;
}
}
if(found)
printf("value %d found at position %d", num, i+1); }.

else {
printf("value %d not found", num);
return 0;
}
```

Output: Enter number of elements: 3
Enter elements: 4 7 5
Enter value to search = 5
value 5 found at position 3

Section - B

(1) find out lenght of a string:

(A)
```
#include <stdio.h>
int main()
int num()
{
    int i, length = 0;
    char str[100];

    printf("Enter a string:");
    gets(str);

    for(i=0; str[i]!='\n'; i++)
        length++;

    printf("Lenght of the string = %d",
    length);
    return 0;
}
```

⇒ output:

Enter a string : parth

Lenght of the string = 5

(2) Convert a string to lower case.

(A)
```
#include<stdio.h>
#include<string.h>
```

```
{
int main()
{
char str[100];
int i;
printf("Enter a string:");
gets(str);
printf("lower case string:%s",
str);
return 0;
}
```

=) output:-

Enter a string: HATHILA

Lowercase string: hathila

(3) ; Convert a string to uppercase.

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

int main()
{
char str[100];
int i;

Printf("Enter a string");
gets(str);
for(i=0; str[i]!='\0'; i++)
{
```

printf("Uppercase string = %s", str);
return 0;
}

⇒ Output:
Enter a string: world
Uppercase string: WORLD.

(4) Convert a string to to glo case.

(A)
```
#include <stdio.h>
#include <string.h>
int main()
{
char str[100];
int i;
printf("Enter a string:");
gets(str);
for(i=0;str[i]!='\n';i++)
{
str[i]=str[i]+32;
}
else if(str[i]>='a' && str[i]<='
{ str[i]=str[i]-32;}
}
printf("Toggle case string:%s",
str);return 0;
}
```

⇒ Output :- Enter a string :- HELLOSIR
Toggle case string :- hello SIR.

(5) Copy one string to another

(A)
```
#include<stdio.h>
#include<string.h>
int main()
{
    char str1[100], str2[100];
    int i;
    printf("Enter a string: ");
    gets(str1);

    for(i=0; str1[i]!='\0'; i++)
    {
        str2[i] = str1[i];
    }
    str2[i] = '\0';
    printf("Copied string: %s", str2);
    return 0;
}
```

⇒ output :- Enter a string :- VIVID
copied string :- VIVID

(6) Compare two strings lexicographically and print which one is greater smaller or same

(A)
```
#include <stdio.h>
int main()
{
  char str1[100], str2[100];
  int i;

  printf("Enter first string: ");
  gets(str1);

  printf("Both strings are same");
  else if(str1[i] > str2[i])
    printf("first string is smaller.");

  return 0;
}
```

→ output : Enter first string : run
Enter second string : stop

first string is SMALLER.

(Q) Reverse a string

(A)
```
#include <stdio.h>
#include <string.h>
int main()
```

```
char str[100], temp;
int i, j;
printf("Enter a string: ");
gets(str);
while(str[j] != '\0')
{
    j++;
}
j = j - 1;
for(i = 0; i < j; i++, j--)
{
    temp = str[i];
    str[i] = str[j];
    str[j] = temp;
}
printf("Reversed string: %s", str);
return 0;
}
```

⇒ output : Enter a string hello

Reversed string olleh

(8) check whether a string is palindrome

(A)
```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];
    int i,j,j;
printf("Enter a string");
gets(str);
j = strlen(str-1);
{
if (str[i] != str[j])
{
printf("palindrome");
return 0;
}
```

→ output : Enter a string = sum

Not a palindrome.

(9) Concatenate one string at the end of another string

(A)
```
#include <stdio.h>
#include <string.h>
int main()
{
```

```
char str1[100], str2[100];
printf("Enter first string:");
gets(str2);

struct(str1, str2);
printf("Concatenated string
        :%s", str1);

return 0;
}
```

=> output:

Enter first string: Enter

Enter second string: name

Concatenated string = entername

(12) print reversed string vertically char by character

```
#include<stdio.h>
int main()
{
    char str[100];
    int i, len;

    printf("Enter a string:");
    gets(str);
```

```
    printf("%c\n", str[i]);
}
return 0;
}
```

⇒ Output : Enter a string : home

e

m

o

h