

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 values: ");
    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, 35, 24
 32, 14

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

(2) Accept 10 values and print 4th, 7th and 9th value (without func)

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

(2) output:

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

4th value = 72, 61

7th value = 45

9th value = 35

(3) Accept 10 values and sort them in ascending / descending order. Find out different techniques for sorting and identify the best ones.

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

 int a[5][5], i, j, temp; //
 // input type

 printf("Enter 5 value: ");

 for (i=0, j<5; j++) { i + n;

 // output of first row

 scanf("%d", &a[i][j]);

}

 for (i=0, j<5, i++) { //

 for (j=i+1, j<5, j++)

 // if a[i][j] = a[i][j+1] then swap

 if (a[i][j] > a[i][j+1]) { //

 temp = a[i][j];

 a[i][j] = a[i][j+1];

 a[i][j+1] = temp;

y

 // two two (

 } // printf(" Ascending order: ");

 for (i=0; i<5; i++)

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

 printf(" In Descending order: ");

 for (i=4, j=0, i--)

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

 printf(" Output of bubble sort is: ");

 printf(" Output of selection sort is: ");

2) Output five numbers with ascending & descending.

Enter 5 values: 33, 56, 98, 67, 13

Ascending order: 13, 33, 56, 67, 98

Descending order: 98, 67, 56, 33, 13

(b) Print minimum number of notes required

(A) #include <iostream.h> // blif (2)
int main() { int i; for (i = 0; i < 9; i++)

int amount, i; count [i] = 0; } (A)

int notes [] = {5000, 2000, 1000, 500, 200, 100, 50, 20, 10};

point & f ("Enter the amount: ");

scanf ("%d", & amount);

for (i = 0; i < 9; i++)

{ if (amount / notes[i] != 0) { amount = amount % notes[i]; }

count [i] = amount / notes [i];

amount = amount % notes [i];

y

point & f ("Minimum notes required: %d\n");

for (i = 0; i < 9; i++) {

point & f ("%d * %d \n", count [i],

notes [i]); } o = 1; } else

else { y = 1; cout << "Invalid number"; }

return 0; }

=> Output: Enter the amount : 28
Minimum notes required = 7 notes

Notes required = 7 notes

1 * 20 = 20 + 5 = 25
1 * 5 = 5

1 * 2 = 2 remaining notes = 1
1 * 1 = 1

20

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

(A) ~~#include <stdio.h>~~ #include <conio.h>
int main() {
 int i, j, sum;
 for (i = 0, j = 0; i < 2, j < 2; i++)
 sum = a[i][j] + b[i][j];
 printf("Sum = %d", sum);
}

point f ("Enter first 4 value : \n");
for (i=0, j<2; i++)

scanf("%d", &a[i][j]);

for (j=0; j<2; j++)

scanf("%d", &b[i][j]);

printf ("Enter second 4 value : \n");
for (i=0, j<2; i++)

scanf("%d", &c[i][j]);

scanf ("%d", &c[i][j]);

for (i=0; i<2; i++)

d

for (j=0; j<2; j++) { }

Sum[i][j] = a[i][j] + b[i][j];

y
pointf("%d", sum[i][j]);

y
printf("%d", sum[i][j]);

return 0;

=> output =

B Enter first value: 2 3 4 5

Enter second value: 9 2 6 5 4

Sum of first and second = 11

9 2 3 4 5 6 7 8 9 10 11 12

(b) Multiply two 2D Arrays and store the result in the 3rd one.

(A) #include <stdio.h>

```
#include <iostream.h>
int main() {
    int a[2][2], b[2][2], m[2][2];
    int i, j, k;
    cout << "Enter first 4 values = ";
    for (i=0; i<2; i++)
        for (j=0; j<2; j++)
            cin << a[i][j];
    cout << "Enter second 4 values = ";
    for (i=0; i<2; i++)
        for (j=0; j<2; j++)
            cin << b[i][j];
    for (i=0; i<2; i++) {
        for (j=0; j<2; j++) {
            m[i][j] = 0;
            for (k=0; k<2; k++)
                m[i][j] += a[i][k] * b[k][j];
        }
    }
    cout << "Resultant matrix = ";
    for (i=0; i<2; i++) {
        for (j=0; j<2; j++)
            cout << m[i][j] << " ";
        cout << endl;
    }
}
```

Scantf("%d", &a[i][j]);

printf("Enter second value: %n");
for (i=0; i<2; i++)

for (j=0; j<2; j++) { for (k=0; k<2; k++)

Scantf("%d", &b[i][j][k]);

for (k=0; k<2; k++)

scanf("%d", &c[i][j][k]);

printf("Multiplication of first and
second = %n");

for (j=0; j<2; j++) { for (i=0; i<2; i++)

printf("%d", mul(i, j));

printf("\n");

Point 3: print "Multiplication of first and second:
returns 0 if -ve or +ve result
y

⇒ Output: Enter first 4 values: 4 5 3 4

Enter second 4 values: 3 6 1 5

Multiplication of first and second:

"%d" showing 49 + 36 = 85 as 85 is +ve result

41 * 36 = 85 as 85 is +ve result

Scan & C[0]d[1], & a[i][j][k]

1st value = 1000 & 1000 and.

printf ("Enter second value: \n")
for (i=0; i<2; i++)

for (j=0; j<2; j++) & for (k=0;

Scan & C[0][0] & 8 b[i][j][k]

for (k=0; k<2; k++)

2. Enter first 4 values

3. printf ("Multiplication of first and
second = \n")

for (i=0; i<2; i++) with for (j=0;

printf ("0d", mult[i][j]);

printf ("\n");

4. Output: 1. 8-bit integer (2)
octal numbers shift left + 128 = 97
y

(1). 8-bit integer (2)

⇒ Output = Enter first 4 values = 4 5 3 4

Enter second 4 values = 3 6 7 5

5. Multiplication of first and second:

1. 8-bit integer (2) + 128 = 1000

2. 8-bit integer (2) + 128 = 1000

(2) Obtain transpose of a 4×4 matrix.

#include < stdio.h>

int main()

{

int i, j, matrix[4][4], transpose[4][4];
printf ("Enter 16 values in ");

for (i=0, j=0; i<4, j++)

{

for (j=0, i<4, j++)

{

scanf ("%d", &matrix[i][j]);

y

for (i=0, j=0; i<4, j++)

{

for (j=0, i<4, j++)

{

transpose[i][j] = matrix[j][i];

y

for (i=0, j=0; i<4, j++)

{

for (j=0, i<4, j++)

{

printf ("%d ", transpose[i][j]);

y

return 0;

y

Output: Enter 16 values:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

Transpose of the matrix = transpose

4 9 6 7 transpose of
 7 6 7 9 transpose of
 5 7 7 9 transpose of

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

```
#include <stdio.h>
#include <limits.h> // for INT_MAX
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<5, i++)
    all[i] = source[i];
printf ("After copying 5 Elements : \n");
for (i=0, i<5, i++)
    printf ("%d ", all[i]);
return 0;
```

⇒ Output : Enter 5 elements of array :
 4 3 5 2 7 after copying 5 Elements :
 0 4 3 5 2 7 0 0 0 0

(9) Reverse an array of maximum 5 elements

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

```
int main()
```

```
{ int i, n, arr[5], j, temp;
```

```
printf("Enter 5 elements\n");
```

```
scanf("%d %d %d %d %d", &n, &arr[0], &arr[1], &arr[2], &arr[3]);
```

printf("Enter 5 elements\n");

```
for (i=0, j=4; i<5, i++)
```

```
{
```

```
scanf("%d", &arr[j]);
```

```
y
```

printf("Reversed array = ");

for (i=0, j=4; i<5, i++)

```
{
```

```
printf("%d", arr[i]);
```

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

```
return 0;
```

```
y (0 = 1 2 3 4 5)
```

→ 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 (ci=0, i<10, i++) {
    for (cj=0, j<10, j++) {
        if (arr[ci] == arr[cj]) {
            count++;
        }
    }
    if (freq[ci] == 0) {
        freq[ci] = count;
    } else {
        freq[ci] += count;
    }
    if (freq[ci] == 0) {
        cout << arr[ci];
    } else {
        cout << arr[ci] << " occurs " << freq[ci] << " times ";
    }
}
return 0;
}

Output: Enter 10 elements: 5 4 6 2 8 7 1 6 1
freq = Frequency of each element

```

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

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

#include <stdio.h>

int main()

int , n, i, choice, temp[10] = {0},
arr[10] = {0};

printf("Enter 10 elements = ");

for (i=0; i<10; i++)

scanf("%d", &arr[i]);

scanf("%d", &choice);

if (choice == 1)

{
 for (i=0, j=9; i<9; i++, j--)

temp[i] = arr[j];

arr[j] = 0;

arr[i] = 0;

```

else if (choice == 2)
{
    for (i = n; i < 10; i++)
    {
        temp[i] = arr[i - n];
    }
    cout << "Shifted array = ";
    pointf ("shifted array = %n");
    return 0;
}

```

(cont'd) after (i = 0, i < n, i++) also title (n) continuing 2 iterations of to previous shifting
pointf ("In")
return 0; // shifts & insertion is
done

\Rightarrow Output of program is

Enter 10 elements

3 5 7 4 5 3 8 9 7 6

(cont'd) insertion of extra 7 after

Enter number of positions to shift
shift direction (1 for left 2 for right)

shift array = 0 0 0 0 0 0 0 0 7 6

2 (i.e. 2nd shift)

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

(i = 0, i < n, i++)

#include < stdio.h >

int main()
{
 int n, newNum, curr, c[11];

cout << "Enter %d elements = %n", n;
for (i = 0; i < n; i++)
{
 cin << curr;
 c[i] = curr;
}

for (i=n; i>0; i--) { // for loop
 ↓
 insertion (array[i-1], element); } // for loop

array[i] = array[i-1]; // for loop
 ↓
 "Element added" is not

array[0] = new1num;

array[0] = new1Num; i = i+1; }
 n++;

{ printf("%d", array[i]); }
 ↓

printf("In %d array = %d - %d\n");
 ↓
 i++;

return 0; // main function

y (if condition is not true)

⇒ Output (in "main") Function

Enter of elements in array is 4

enter 4 elements : 3 5 2 7

Enter new number at first place.

It is used to replace first element with new one.

• Array after insertion is 3 3 5 2 7.

↓
 3 3 5 2 7 → 3 3 5 2 7

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

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

int main() { // for testing

int i, n, pos, arr[20]; i = 120;

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

printf ("Enter elements: ");

for printf ("Enter new number: ");
scanf ("%d", &num);

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

}

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

$$a[pos-1] = num;$$

printf ("New array: ");

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

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

return 0;

y F.S. is 2nd element is 3rd pos

=> Output: Enter number of elements

5

Enter elements: 5 8 6 3

Enter position: 2

Enter new number: 9

NA

New array: 5 9 8 6 3 which is 5 9 8 6 3

(iv)

Insert a new number at the first position of an array.

(A)

Includes stdio.h + min + max
int main () {

Ques 2. Inserting and deleting elements.

```

int i, n, num;
char arr[20];
printf("Enter the array elements: ");
scanf("%d", &n);
for (i = 0; i < n; i++)
    printf("%d ", arr[i]);
printf("\nEnter the number of elements to insert: ");
scanf("%d", &num);
printf("Enter new number to insert at last: ");
scanf("%d", &num);
    
```

arr[n] = num;

n++;

to print arr[i] for i = 0 to n-1

for (i = 0, i < n; i++)
 printf("%d", arr[i]);

else return output after the loop.

y

Output: Enter the 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 (A)

(continues on next page)

Delete value from the first position
of an array is (a) 1. f(m)
(b) 2. f(m-1)

#include < stdio.h > // header file

int main()
{

int i, n, num;

printf("Enter the array elements: ");
 scanf("%d", &n);

```

printf("Enter elements: ");
for (i=0; i<n; i++) {
    // scan & store 'd' & a[i]
}

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

return 0;

```

⇒ Output :
 Enter elements = 4
 Enter elements = 27 5 7
 Array after deleting first element
 7 5 7

(16) Deletes 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++)
        a[i] = a[i+1];
}

```

```

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

```

return 0;

Output: Enter number of elements = 4

Enter elements: 6 14 8 2
 Enter position to delete = 3

(7) Delete an 'a' value from the last position of an array.

(A) #include <iostream.h>
 int main()
 {

int a[4]; int i;
 cout << "Enter elements: ";

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

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

pointf("\n");
 return 0;
}

Explanation: After printing the array, the user has to enter the number of elements.

Output: Enter number of elements:

Enter elements: 6 4 11 8 2 5

Array After deleting last elements

6 4 11 8 2

(18) Delete a value from other arrays.

(A) #include <stdio.h>

#include <conio.h> // for clrscr()

{

int i, j, n, num, arr[20];

clrscr(); // for clearing screen

printf("Enter number of elements");

scanf("%d", &n);

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

scanf("%d", &arr[i]);

printf("New array: ");

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

printf("%d ", arr[j]);

j++;

=> ~~order of output~~ (answering to question)

Enter number of elements: 5

Enter elements: 6 4 9 7 11

Enter value to delete: 9

New array: 6 4 7

elements to remove using function

(19) Search a value within an array

#include < stdio.h> // header file
int main ()

```

int i, n, num; // declaration of variables
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;
    }
}

```

else if (found == 1) {
 printf ("Value %d found at position %d", num, i+1);
}

else {

printf ("Value %d not found", num);
 return 0;

} // end of main function

Output: Enter number of elements : 3
Enter elements : 4 5 6

Enter value to search : 5

Value found at position 2

Section - B

Topic - Input & Output using Library

(1) find out length of a string:

(A) #include <stdio.h>

#include <limits.h> // for INT_MAX

int main() { int length = 0; char str[100];

scanf("%s", str); // reading string

length = strlen(str); // length of string

char str[100] ; length = strlen(str);

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

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

printf("length of the string = %d\n", length);

⇒ output :

Enter a string: path

Length of the string = 5

(2) Convert a string to lower case.

result = result - 'A' + 'a'; // ASCII value of A = 65, a = 97

(A) #include <stdio.h>

#include <string.h>

Condition to break loop

```

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

```

(A)

=) output :-
 Enter a string : ~~PATIL~~ HATHIL
 Lowercase string: ~~patil~~ huthila A

(B) Conversion of string to uppercase.

```

#include <stdio.h>

```

```

int main()
{
    char str[100];
    int i;
    gets(str);
    for(i=0; str[i] != '\0'; i++)
        str[i] = str[i] - 'a' + 'A';
    printf("Upper case string = %s", str);
}
```

```

    int i;
    char str[100];
    int i;
    gets(str);
    for(i=0; str[i] != '\0'; i++)
        str[i] = str[i] - 'a' + 'A';
    printf("Upper case string = %s", str);
}
```

Print of (B) Enter a string: " "
 gets(str);
 for (i=0, str[i] = ' ', i++)

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

⇒ Output: C:\Users\DELL\PycharmProjects\PythonProject> python3.8 upper.py
 Enter a string: siva

Uppercase string: SIVA.

(ii) Convert a string To to uppercase.

(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] != '\0'; i++)

if (str[i] >= 'a' & str[i] <= 'z')

else if (str[i] >= 'A' & str[i] <= 'Z')
 { str[i] = str[i] - 32; }

printf("Toggled string is: %s", str);
 return 0;

Q) Output :- Enter a string : HELLOSir
 program is doing nothing here
 Toggle case of string & helloSIR.

Exhibit 2.2 Standard I/O (A)

(5) Copy one string to another

(A) ~~#include <std.h>~~ main()

#include <string.h> i + m
 int main ()

char str1[100], str2[100];
 l

char str1[100], str2[100];

int i, j; str1[i] = str2[j]

for (i = 0, j = 0; str1[i] != '\0'; i++)
 gets(str1);

str2[i] = str1[j];

if (str1[i] == 'A' - 'a') str1[i] = 'T' + 'a' - 'A';

if (str1[i] == 'E' - 'a') str1[i] = 'R' + 'a' - 'E';

return 0; } // infinite loop

Output :- Enter a string : VIVID
 copied string part : VIVID

Exhibit 2.2 Standard I/O (A)

Exhibit 2.2 Standard I/O (A)

CD number 100

(6)

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

(A)

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

```
int main()
{
```

```
    char c1[100], s1[100];
    int i, j;
    cout << "Enter first string: ";
    cin << s1;
    cout << "Enter second string: ";
    cin << c1;
```

```
    printf("Enter first string: ");
    gets(c1);
    printf("Enter second string: ");
    gets(s1);
```

```
    if (c1 > s1)
        printf("Second string is greater");
    else if (c1 < s1)
        printf("First string is greater");
    else
        printf("Both strings are same");
```

```
    return 0;
}
```

→

Output : Enter first string : sun
Enter second string : stop
first string is SMALLER.

first string is SMALLER.

(7)

Reverse a string : Using no

(A)

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

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

char str[100], temp;

int i, j; // i=0, j=0

printf ("Enter a string: ");

getstr(str); // input

while (str[j] != '\0')

{ // j++ is printing // output

j = j - 1;

for (i = 0; i < j, i++, j--) {

{ // i++ is printing // output

temp = str[i];

str[i] = str[j];

// printing

}

printf ("Reversed string: %s", str);

now returning to original string

y

overwriting o + o

∴ output : Enter a string to reverse

Reversed string is (auto)

using goto to break

(d) output given is (A)

equivalent of this is (B)

or natural language

(8)

check whether a string is palindrome

(A)

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

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

```
char str[100];  
int i, j;
```

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

```
j = strlen(str) - 1;
```

```
for (i = 0; i <= j; i++) {
```

```
if (str[i] != str[j]) {
```

```
printf("Not a palindrome");
```

```
return 0;
```

```
}
```

Output : Enter a string sum

Not a palindrome.

(9)

Concatenate two strings at the end of another string

(A)

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

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

int main()

char str1[100], str2[100];
 printf ("Enter first string\n");
 printf ("Enter second string\n");
 gets(str1);
 gets(str2);

struct strct {str1, str2};

mon = struct {str1 + str2};
 printf ("Concatenated string
 : %s ".strct);

return 0;

3

=> output:

Enter first string & Enter
 Enter. second string - name

Concatenated string = entername

(12) print reversed string vertically character by character

#include <stdio.h>
 int main ()

char str[100];
 int i, len;

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

(USING STATE COMPILE WITH PWD)
 printf("1-%c\n", str[i]);
 return 0;

→ Output : Enter a string "home"
 home

m

o

h

computer is

E

also find a useful file exit satya

on this project b noise . setting

instructions - input1, b9b9b9 (0)

then eliminate after backslash from
 continuation part

C:\01\file > abulbaba

C:\minicom

C:\001>+2 serial

English, 1 + 1

C:\001>+2 serial 25600) A fiction