CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

EXPERIMENT NUMBER –

TOPIC OF EXPERIMENT –

Practical 8.1: WAP to read an array of elements and print the same in the reverse order along with their addresses using pointer.

Practical 8.2: Write a function code that is returning pointer to the larger value out of two passed values.

Practical 8.3: The bank balance of N persons of a city is recorded. Due to COVID-19 government has decided to credit accounts with Rs. 1000 of all those persons whose balance is nil, write a function Update balance () by passing pointer to an array as argument and print the updated bank balance list in calling function.

Practical 8.4: The CGPA of 5 semesters of N students has stored in NX5 array and names of corresponding students are stored in separate string array The student who got average CGPA >=8 is eligible for placements. Pass pointers to both arrays to function not eligible () and print list of non-eligible students.

AIM OF THE EXPERIMENT – Problem solving using pointer

1.

Algorithm:

1. Declare all the variable like n,I,array;
2. Declare pointer for swapping/reversing number.
3. Declare array using pointer.
4. Use for loop for iterating array for inserting number.
5. Print inserted element which is initially inserted in arrays.
6. Now print reverse of the element.
7. Use for loop for printing reverse of the element which is initially inserted.

Code:

```
#include <stdio.h>

int main()

{

 int n, i, arr1[15];

 int *pt;

 printf("\n\n Pointer : Print the elements of an array in reverse order :\n");

 printf("----------\n");

 printf(" Input the number of elements to store in the array (max 15) :\n ");

 scanf("%d",&n);

  pt = &arr1[0]; // pt stores the address of base array arr1

 printf(" Input %d number of elements in the array : \n",n);

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

 {

         printf(" element - %d : ",i+1);

         scanf("%d",pt);//accept the address of the value

         pt++;

        }

          pt = &arr1[n - 1];

 printf("\n The elements of array in reverse order are :");

 for (i = n; i > 0; i--)

 {

   printf("\n element - %d : %d ", i, *pt);

   pt--;

 }
```

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

  return 0;

}
```

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered):

Actually, I was not getting any error but I get warning by compiler because I take input by user as integer by %s but this is not perfect right so I get error but I fix this error by %d in place of %s.

Output:

2.2.

```
Pointer : Print the elements of an array in reverse order :
------------------------------------------------------------------
Input the number of elements to store in the array (max 15) :
5
Input 5 number of elements in the array :
element - 1 : 1
element - 2 : 2
element - 3 : 3
element - 4 : 4
element - 5 : 5

The elements of array in reverse order are :
element - 5 : 5
element - 4 : 4
element - 3 : 3
element - 2 : 2
element - 1 : 1
```

2.

Algorithm:

1. Declare all variable.
2. Now all variable assign by pointer for comparision.

3. Use if else condition.

4. Declare function for finding maximum using pointer.

5. Find using max(x,y);

6. Now call the function for finding maximum of two number.

Code:

```
#include <stdio.h>

int *getMax(int *, int *); //function declaration

int main(void) {

 // integer variables

 int x, y;

 printf(" FIND MAX VALUE\n Enter two integers-\n ");

 scanf("%d%d", &x,&y);

 // pointer variable

 int *max = NULL;

  /**

  * get the variable address that holds the greater value

  * for this we are passing the address of x and y

  * to the function getMax()

  */

 max = getMax(&x, &y);

  // print the greater value

 printf(" Max value: %d\n", *max);
```

```
 return 0;
}
// function definition
int *getMax(int *m, int *n) {
 /**
  * if the value pointed by pointer m is greater than n
  * then, return the address stored in the pointer variable m
  */
 if (*m > *n) {
   return m;
 }
 /**
  * else return the address stored in the pointer variable n
  */
 else {
   return n;
 }
}
```

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered):

Actually, I was not getting any error but I get warning by compiler because I take input by user as integer by %s but this is not perfect right so I get error but I fix this error by %d in place of %s.

Output:

```
FIND MAX VALUE
Enter two integers-
50 150
Max value: 150

--------------------------------
Process exited after 4.486 seconds with return value 0
Press any key to continue . . .
```

3.

Algorithm:

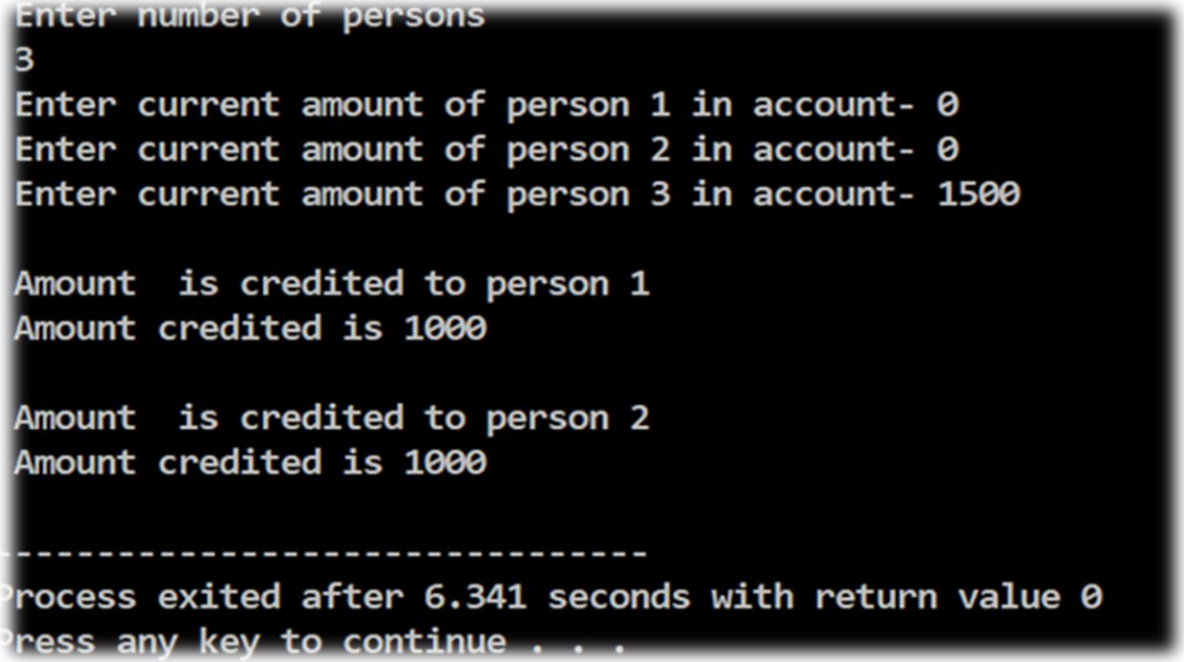
1. Declare all variables.
2. Declare arrays to store number of people.
3. Now declare variables using pointer for assigning people who have to null account.
4. Take input data from user.
5. Now use for loop for iterating all the people.
6. Now declare function for update_balance.
7. Now print.

ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION

(Kindly jot down the compile time errors encountered):

Actually, I was not getting any error but I get warning by compiler because I take input by user as integer by %s but this is not perfect right so I get error but I fix this error by %d in place of %s.

Output:



4.

Algorithm:

1. Declare all variables.

2. Take all the input from user.

Code:

```
#include<stdio.h>

struct sp

{

        char name[100];

}s[100];
```

```
int eligible(double b,struct sp s[],int k)
{
        printf("\nstudent : %s and c.g.p.a : %0.2lf elgible\n",s[k].name,b);
}

int not_eligible(double b,struct sp s[],int k)
{
        printf("\nstudent : %s and c.g.p.a : %0.2lf not elgible\n",s[k].name,b);
}
int main()
{

        int n,i,j;
        double a[100][5];
        printf(" Enter number of students- ");
        scanf("%d",&n);
        double sum[n],cgpa[n];
        for(i=0;i<n;i++)
        {
                printf(" %d. Enter student name- ",i+1);
                scanf("%s",&s[i].name);
                for(j=0;j<5;j++)
                {
                        printf(" Enter CGPA (0-10) in sem %d- ",j+1);
                        scanf("%lf", &a[i][j]);
```

```
                }
        }

        for(i=0;i<n;i++)
        {
                sum[i]=0;
                for(j=0;j<5;j++)
                {
                        sum[i]= sum[i]+ a[i][j];
                }
                cgpa[i]= (double)sum[i]/5.0;

                if(cgpa[i] >=8)
                {
                        eligible(cgpa[i],s,i);
                }
                else
                {
                        not_eligible(cgpa[i],s,i);
                }
        }
        return 0;
}
```

Output:

```
 Enter number of students- 2
 1. Enter student name- Ramandeep
 Enter CGPA (0-10) in sem 1- 8.5
 Enter CGPA (0-10) in sem 2- 8.6
 Enter CGPA (0-10) in sem 3- 8.2
 Enter CGPA (0-10) in sem 4- 8.5
 Enter CGPA (0-10) in sem 5- 8.6
 2. Enter student name- Shruti
 Enter CGPA (0-10) in sem 1- 8.2
 Enter CGPA (0-10) in sem 2- 8
 Enter CGPA (0-10) in sem 3- 7
 Enter CGPA (0-10) in sem 4- 7.9
 Enter CGPA (0-10) in sem 5- 7.8

student : Ramandeep and c.g.p.a : 8.48 elgible

student : Shruti and c.g.p.a : 7.78 not elgible

--------------------------------
Process exited after 58.12 seconds with return value 0
Press any key to continue . . .
```