

Name : Yana Srivastava

Uid : 20BCS2279

Section : 23

Group : “B”

Subject : Problem Solving With Programming Lab

Experiment : 8

Topic :

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

Algorithm :

1. Start the program.
2. Declaration of variables in integer datatype.
3. Declaration of pointer to the integer.
4. Print the statement and accept the input of number of elements of an array from the user.
5. Stores the address of the base array.
6. Print the number of elements of an array and execute the for loop to enter it.

7. Print the statement and accept the input of elements from the user.
8. Print the statement and execute the for loop to enter the elements in reverse order.
9. Print the elements in reverse order.
10. End the program by returning an integer.

Code :

```
//creating a header file for standard input output operations
#include<stdio.h>

//function which returns integer value
int main()
{
//declaration of variables in integer datatype
    int i,n,arr[20];

    //declaration of pointer to the integer
    int *p;

//print the statement
    printf("Input the number of elements:\n");

//accept the input of number of elements of an array from
the user
```

```
scanf("%d",&n);  
//stores the address of the base array  
p=&arr[0];  
//print the number of elements of an array  
printf("Input the %d number of elements:\n",n);  
//execute the loop to enter the number of elements  
for(i=0;i<n;i++)  
{  
//print the statement  
printf("Elements-%d:",i+1);  
//accept the input of elements from the user  
scanf("%d",p);  
p++;  
}  
p=&arr[n-1];  
//print the statement  
printf("Elements of array in reverse order:\n");  
//execute the loop to enter the elements in reverse order  
for(i=n;i>0;i--)  
{
```

```
//print the elements in reverse order
printf("\nElement-%d:%d",i,*p);

    p--;
}

printf("\n\n");
//return an integer
return 0;

}
```

Errors encountered during program's execution :

No Error.

Output :

```
Input the number of elements:
6
Input the 6 number of elements:
Elements-1:1
Elements-2:2
Elements-3:3
Elements-4:4
Elements-5:5
Elements-6:6
Elements of array in reverse order:

Element-6:6
Element-5:5
Element-4:4
Element-3:3
Element-2:2
Element-1:1
```

Program's Explanation (In Brief) :

In this program we have to read an array of elements and print the same elements in reverse order along with their addresses using pointer. Pointer is used to store the address of base array.

Topic :

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

Algorithm :

1. Start the program.
2. Function declaration.
3. Function which returns integer value.
4. Declaration of integer variables.
5. Print the message and accept the input of two numbers from the user.
6. Declaration of pointer variables.
7. Passing the address of x and y to the function getMax() to get the variable address that holds the greater value.
8. Print the maximum value.
9. Return an integer.
10. Function definition.
11. Check using if else statement if the value pointed by pointer m is greater than n then, return the address stored in the pointer variable m else return the address stored in the pointer variable n.
12. End the program.

Code :

```
//creating a header file for standard input output
operations
#include <stdio.h>

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

//function which returns integer value
int main()
{
//declaration of integer variables
int x,y;

//print the message
printf("Enter the two numbers:");

//accept the input of two numbers from the user
scanf("%d%d",&x,&y);

//declaration of pointer variables
int *max = NULL;

/*passing the address of x and y to the function getMax()
to get the variable address that holds the greater value*/
max = getMax(&x, &y);
```

```
//print the maximum value
printf("Max value: %d\n", *max);
//return an integer
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 :

No Error.

Output :

```
Enter the two numbers:10
12
Max value: 12
```

Program's Explanation (In Brief) :

In this program we have to create a function which returns the maximum value using pointer and check the condition using if else conditional statement.

Topic :

The bank balance of N persons of a city are recorded. Due to COVID-19 government has decided to credit accounts with rupees 1000 of all those persons whose balance is NULL , Write a function Update_balace() by passing pointer to an array as argument and print the updated bank balance list in calling function.

Algorithm :

1. Start the program.
2. Function declaration.
3. Array to store the credited amount.
4. Print the message and credited amount.
5. Function to return an integer.
6. Declaration of array and variables in integer datatype.
7. Stores the base address of array
8. Execute the loop to enter the number of persons.
9. Print the statement and accept the input from the user.
10. Check using conditional statement if the current bank balance is null or not and then print the updated bank balance according to it.
11. End the program by returning an integer value.

Code :

```
//creating a header file for standard input output operations
#include<stdio.h>

//function declaration
int Update_balance(int *p, int a[], int i)
{
//array to store the credited amount
p= a[i];
//print the message
printf(" Amount credited is %d\n", p);
}
```

```
//function which returns integer value
int main()
{
//array which store the number of persons
int a[100];
//declaration of variables in integer datatype
int n,i,*p;
//print the message
printf(" Enter number of persons\n ");
//accept the input from the user
scanf("%d", &n);
//stores the address of base array
p= &a[0];
//execute the loop to enter the number of persons
for(i=0;i<n;i++)
{
//print the message and accept the current amount of person
from the user
printf(" Enter current amount of person %d in account- ",i+1);
scanf("%d",p);
    p++;
}
```

```

    }
//stores the address of base array
p = &a[0];
for (i=0;i<n;i++)
{
if(*p==0)
    {
*p = *p + 1000;
//print the message and updated balance
printf("\n Amount is credited to person %d \n", i+1);
Update_balance(&p,a,i);
    }
    p++;
}
//return an integer
return 0;
}

```

Errors encountered during program's execution :

No Error.

Output :

```
Enter number of persons
4
Enter current amount of person 1 in account- 0
Enter current amount of person 2 in account- 0
Enter current amount of person 3 in account- 0
Enter current amount of person 4 in account- 10

Amount is credited to person 1
Amount credited is 1000

Amount is credited to person 2
Amount credited is 1000

Amount is credited to person 3
Amount credited is 1000
```

Program's explanation (In Brief) :

In this program we have to create a function `Update_balance()` by passing pointer to an array as argument and print the updated bank balance list in calling function. And by using conditional statement we have to check that the current bank balance of a person is null or not and print the updated bank balance according to it.

Topic :

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 array to function `not_eligible()` and print list of non-eligible students.

Algorithm :

1. Start the program.
2. Declaration of variables in character and integer datatype.
3. Function declaration.
4. Check eligibility and print the message.
5. End.

Code :

```
//creating a header file for standard input output operations
#include<stdio.h>

//creating a structure
struct sp
{
//declaration of variables in character and string datatype
char name[100];
}
s[100];

//function declaration
int eligible(double b,struct sp s[],int k)
{
    //print the message
    printf("\nstudent : %s and c.g.p.a : %0.2lf\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  
eligible\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;  
}
```

Errors encountered during program's execution :

No Error.

Output :

```
Enter number of students- 4
1. Enter student name- yana
Enter CGPA (0-10) in sem 1- 8
Enter CGPA (0-10) in sem 2- 8.5
Enter CGPA (0-10) in sem 3- 9
Enter CGPA (0-10) in sem 4- 10
Enter CGPA (0-10) in sem 5- 7
2. Enter student name- ram
Enter CGPA (0-10) in sem 1- 10
Enter CGPA (0-10) in sem 2- 8
Enter CGPA (0-10) in sem 3- 9
Enter CGPA (0-10) in sem 4- 7
Enter CGPA (0-10) in sem 5- 6
3. Enter student name- shyam
Enter CGPA (0-10) in sem 1- 10
Enter CGPA (0-10) in sem 2- 9
Enter CGPA (0-10) in sem 3- 8
Enter CGPA (0-10) in sem 4- 7
Enter CGPA (0-10) in sem 5- 6
4. Enter student name- ravi
Enter CGPA (0-10) in sem 1- 2
Enter CGPA (0-10) in sem 2- 3
Enter CGPA (0-10) in sem 3- 4
Enter CGPA (0-10) in sem 4- 5
Enter CGPA (0-10) in sem 5- 6
```

```
tudent : yana and c.g.p.a : 8.50 eligible
tudent : ram and c.g.p.a : 8.00 eligible
tudent : shyam and c.g.p.a : 8.00 eligible
tudent : ravi and c.g.p.a : 4.00 not eligible
```

Program's Explanation :

In this program we have to input the number of students and their cgpa in 5 semester and Pass pointers to both array to function not_eligible() and print list of non-eligible students.