Covered Topics Under UNIT-5 of "PPS-PROGRAMMING FOR PROBLEM SOLVING (BCS101 / BCS201)"

shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

Shweta Tiwari

Class Notes

Published Date: November, 2022

PPS: UNIT-5

Pointers

File Handling

FALL SEMESTER, YEAR (I/II sem, 1st yr)

FALL SESSION (2022-23)
(PPS)
MS. SHWETA TIWARI
Published Date: November, 2022

shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

By SHWETA TIWARI
Under On: UNIT-5

PREPARED FOR
Engineering Students
All Engineering College

PREPARED BY
SHWETA TIWARI

Units Classified into Chapters

PPS: UNIT-5:

Chapter-12 ENUMERATION
Chapter-13 POINTER

Chapter 12

ENUMERATION

The enumerated data type is used when we know in advance about a finite list of values as a particular data type. The declaration of enumerated data type start with **'enum'** keyword.. **'enum'** keyword is followed by an enum name, an open brace, each of the values separated by a comma, and finally a closing brace and a semicolon. The list of values is known as the enumerated list of items.

Syntax: -

enum <enum name>{value1,value2,value3……….value n};

example:

enum color { red, blue, green, white, black };

Note:- Here color is the name of the enum, and red, green, blue are the symbolic constants. Each enum constant has an integer value. If we have not given any value, then the initial value of the red is 0, blue is 1 and so on and incremented by one.

As shown in the example below:

```
#include<stdio.h>
#include<conio.h>
enum col{red, green, blue, yellow, black, white };
void main()
{
clrscr();
printf("\n the value of red=%d",red);
printf("\n the value of green=%d",green);
printf("\n the value of white=%d",white);
getch();
}
```

output:-

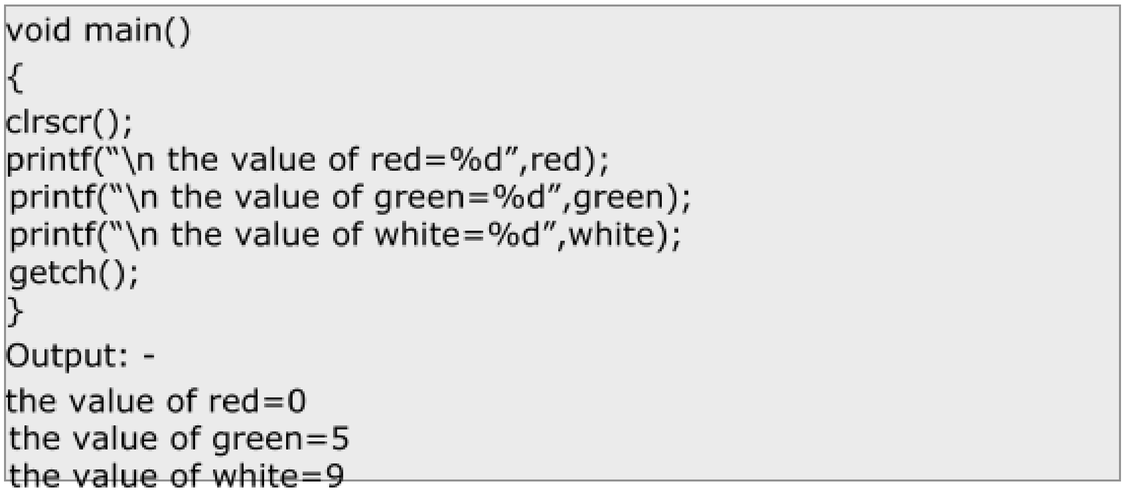
```
the value of red=0
the value of green=1
the value of white=5
```

Initialization of enum constant

We can also change the value of the each constant in the enum. Any constant can be initialized with a particular value, and those that are not initialized ,will be incremented automatically.

As shown in the example below:

```
#include<stdio.h>
#include<conio.h>
enum col{red, green=5, blue, yellow, black, white };
```



```
void main()
{
clrscr();
printf("\n the value of red=%d",red);
 printf("\n the value of green=%d",green);
 printf("\n the value of white=%d",white);
 getch();
}
Output: -
the value of red=0
 the value of green=5
 the value of white=9
```

Important Short Answer Type Questions

1. Give an example of enumerated data type in C.

Important Long Answer Type Questions

1. Explain enum declaration in C with suitable example.
2. How can you differentiate enum from an array?

Chapter 13

POINTER

Pointer is a special type of variable, which is used to store the address of another variable. This variable starts with asterisk symbol (*). The pointer can use two special operators.

1. The first one is *(asterisk), it is called indirection operator or value of operator.
2. The second is &(ampersand), it is called address of operator.

Pointer declaration

A pointer is declared same as a variable with a specific data type. But declared with an asterisk sign. As show in the example below:

Example

```
int *ip;              //ip is our integer
pointer float *fp; //fp is our Float pointer
char *cp;    //cp is our character pointer
```

Address operator & (Ampersand)

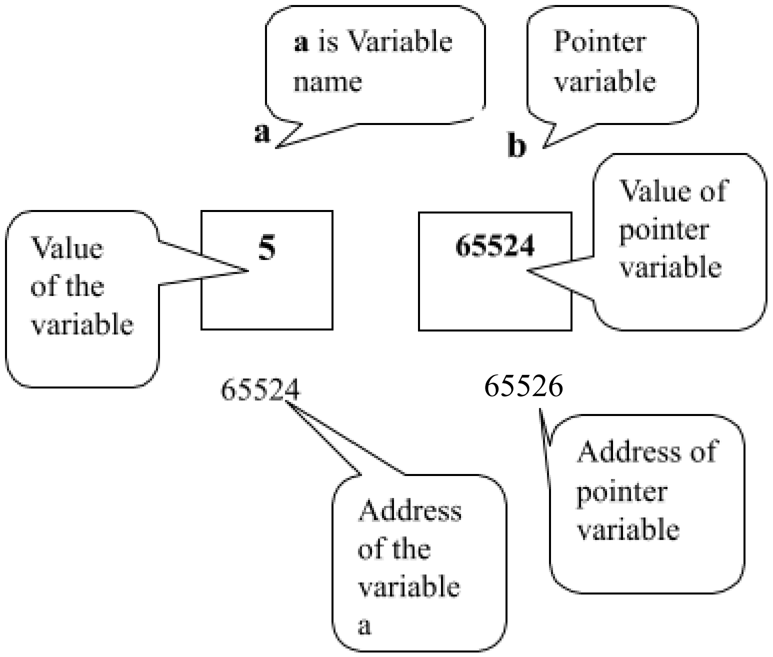
As mentioned above, pointer uses two special operators; one of them is the address operator, which is used to assign the address of the variable to the pointer variable. As shown the example below:

```
int a,*ip;
a=5;
          ip=&a;       /*here we can assign the address of variable a
          to pointer variable ip*/
```

Indirection operator * (asterisk)

Indirection operator is also known as value of operator, which is used to print the value of the variable through pointer variable.The indirection operator is used to declare a pointer. As shown in the example below:

```
int a,*ip;
        a=5;
        ip=&a;
        printf("\n the value of a=%d",a);
  printf("\n the value through pointer=%d",*ip); /*here ip print the
                                                 value of a */
```


a is Variable name
a
Pointer variable
b
Value of the variable
5
65524
Value of pointer variable
65524
65526
Address of the variable a
Address of pointer variable

Complete example of pointer variable

```
# include<stdio.h>
 # include<conio.h>
void main()
{
int a, *b;
clrscr();
a=5;
b=&a;
printf("\n the value of a=%d",a);
printf("\n the address of a=%u",&a);
printf("\n the value of b=%u",b);
printf("\n the value of b through pointer =%d",*b);
getch();
}
output
the value of a=5
the address of a=65524
the value of b=65524
the value through pointer b=5
```

Write a program to add two numbers using pointers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,,b,*c,*d, e;
clrscr();
printf("enter the value of a=");
scanf("%d",&a);
printf("enter the value of b=");
scanf("%d",&b);
c=&a;
d=&b;
e=*c + *d;
printf("sum=%d",e);
getch();
}
```

Write a program to swap two numbers using pointer without using third variable.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a, b, *c, *d;
clrscr();
printf("enter the value of a=");
scanf("%d", &a);
printf("enter the value of b=");
scanf("%d" , &b);
c=&a;
d=&b;
//swapping of two numbers
*c= *c+ *d;
*d= *c - *d;
*c= *c- *d;
printf("\after swapping the numbers");
printf("\n now the value of a=%d",a);
printf("\n now the value of b=%d",b);
getch();
}
```

Arithmetic pointers

We can also use pointers for arithmetic purposes same as variables. When we write a++ it increase the value of the variable but when we write p++, it is a pointer variable it moves the pointer to the next memory location. Like

```
int *p;

p++;

or

p=p+3;
```

Assume that the address of p is 65524.

When we add 3 to the pointer it move to the next location is 65527.

Note that we can perform any arithmetic operation on a pointer variable.

Pointers with array

The values of the array can be accessed with the help of pointer. Normally we can assign the address of first element of an array to the pointer. When we increment inside the pointer, it increments its address. It means it reads the address of next element. The address of first element of an array is called base address of array.

As shown in the example below:

```
    int a[]={2,4,3,5,6};
    int *ip;
    //here we can assign the address of array to the pointer.
    ip=&a[0];
    /*We can also assign the address of the first element to the pointer*/
    ip=a;
  //Here we need not to mention the index of the array.
```

Write a program to find the maximum number from the given array-using pointer.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int a[5],*p, i, b;
clrscr();
for(i=0;i<5;i++)
{
printf("enter any number");
scanf("%d",&a[i]);
}
p=&a[0];
b=a[0];
for(i=0;i<5;i++)
{
if(*p>b)
{
```

```
b=*p;
}
p++;
}
printf("the maximum number=%d",b);
getch();
}
```

Pointer with strings
A string is a group of characters, which contain characters. We can access the string-using pointer that read one character at time. We can assign the address of first element of character array to the pointer.
As shown in the given example below:

Write a program to print your name-using pointer.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char a[50]="Charanjiv Singh";
char *p;
clrscr();
p=a;
while(*p!='\0')
{
printf("%c",*p);
p++;
}
getch();
}
```

Pointer with two Dimensional array

We can also access the two-dimensional array-using pointer in which we can declare array with pointer.

As shown in the program: -

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3][3],*p,i,j;
clrscr();
for(i=0;i<3;i++)
{
```

```
for(j=0;j<3;j++)
{
printf("enter any number");
scanf("%d",&a[i][j]);
}
}
p=&a[0][0];
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
printf("%d",*p);
p++;
}
printf("\n");
}
getch();
}
```

Pointer with structure

We can also access the elements of the structure with help of pointer. We can assign the address of the structure variable to the pointer variable. For accessing the elements we can use arrow (->) operator in place of dot(.) operator.

Write a program to access the elements of the structure-using pointer.

```
#include<stdio.h>
#include<conio.h>
struct stu
{
char name[20];
int roll;
};
void main()
{
struct stu s={"Pratham",1};
struct stu *p;
clrscr();
p=&s;
printf("\n your name=%s",p->name);
printf("\n your roll=%d",p->roll);
getch();
}
```

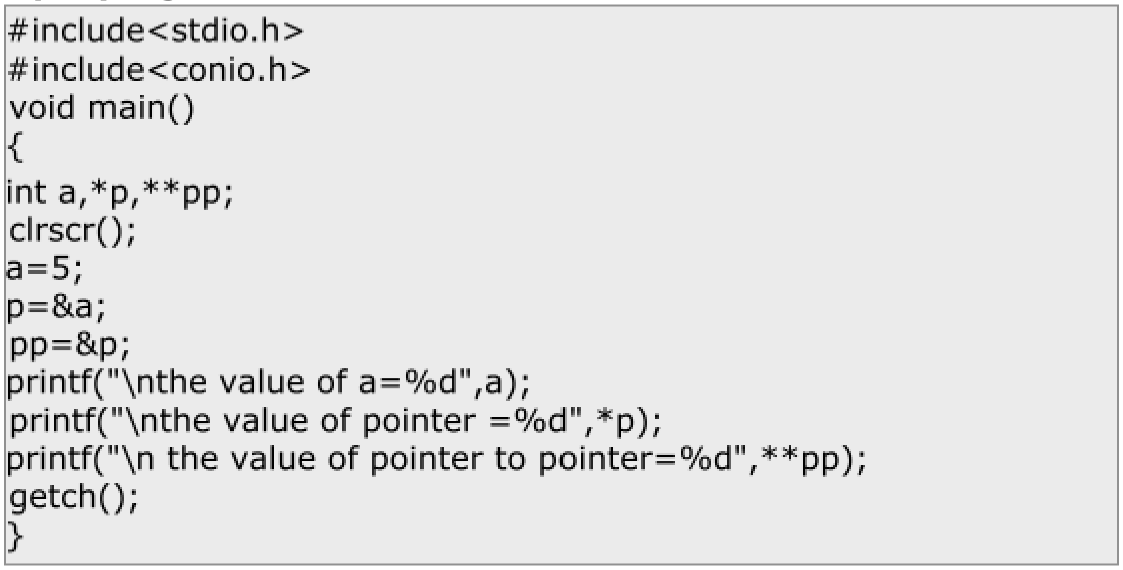
Pointer to pointer

Pointer to pointer means which can store the address of another pointer variable. A pointer-to-pointer variable start with double asterisk (**) sign. A simple pointer can store the address of a variable but a pointer to pointer can store the address of another pointer. As shown in the example below:

Example :-

```
int a, *p,**pp;
a=5;
p=&a;
pp=&p;
```

Example program

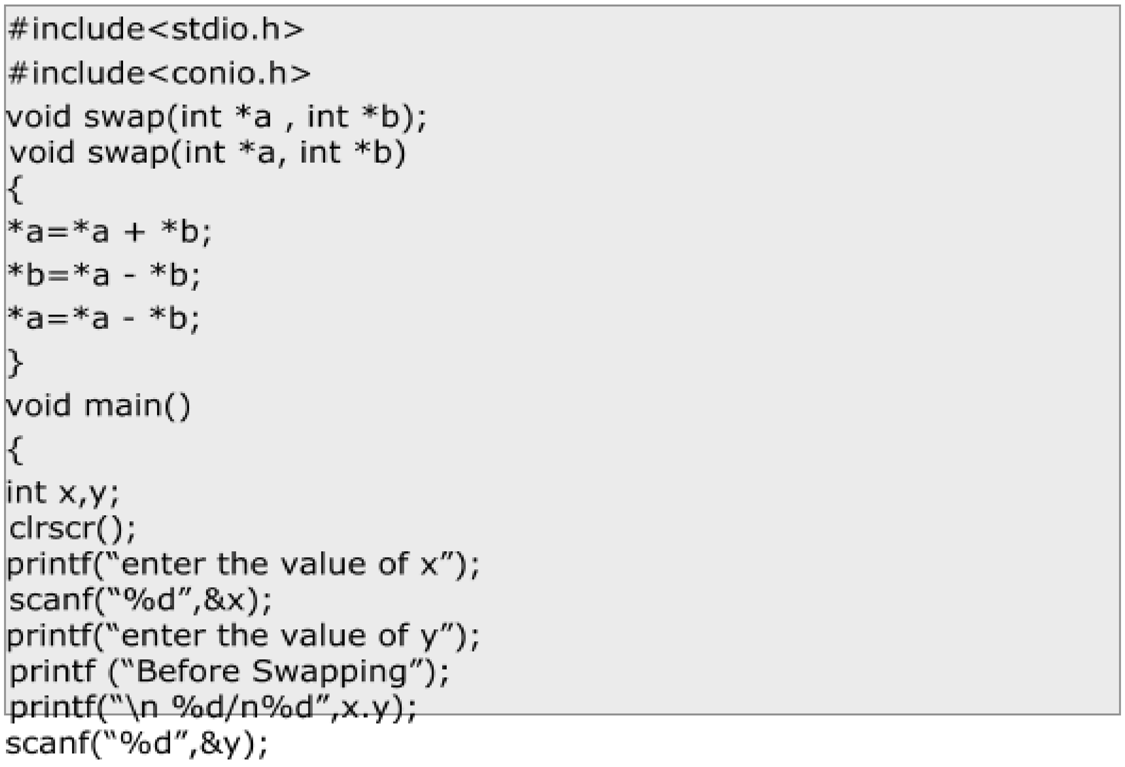


```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,*p,**pp;
clrscr();
a=5;
p=&a;
pp=&p;
printf("\nthe value of a=%d",a);
printf("\nthe value of pointer =%d",*p);
printf("\n the value of pointer to pointer=%d",**pp);
getch();
}
```

Pointer passing as argument to the Function

We can also pass pointer variable as an argument into the function. This is also known as reference variables.

As shown in the example below:



```
#include<stdio.h>
#include<conio.h>
void swap(int *a , int *b);
void swap(int *a, int *b)
{
*a=*a + *b;
*b=*a - *b;
*a=*a - *b;
}
void main()
{
int x,y;
clrscr();
printf("enter the value of x");
scanf("%d",&x);
printf("enter the value of y");
printf ("Before Swapping");
printf("\n %d/n%d",x,y);
scanf("%d",&y);
swap(&x ,&y);
printf("\n after swapping ");
printf("\n the value of x=%d",x);
printf("\n the value of y=%d",y);
getch(); }
```

Function Returning Pointer
A function can return a pointer same as it returns a variable. In this we can declare and define the function with indirection (value of operator) operator.

As shown in the example given below

```
# include<stdio.h>
#include<conio.h>
int *show();
int *show()
{
int a;
printf("enter any number");
scanf("%d", &a);
a=a*a;
return &a;
}
void main()
{
int *p;
clrscr();
p=show();
printf("\n The value of a=%d",*p);
getch();
}
```

Advantages of Pointers

1. Pointers increase the speed of the program.
2. It is used to read the address of the variable.
3. It is used to the manage memory efficiently.
4. For Dynamic memory allocation we can use pointers.
5. With the help of pointers we can create efficient programs.
6. Pointers can be used to access elements of the structure.
7. It can pass as arguments into the function.

Difference Between A Variable And Pointer Variable

Variable	Pointer variable

A variable starts with an alphabet.	A pointer variable starts with an asterisk

	sign.
It can store the value.	It can store the address of another variable
It can not use any special operator.	It uses two special operators first is address operator and second is indirection operator(value of operator).

When we perform any arithmetic operation on variable it increases the value of the variable. Like
int a=5;
a++;
Now value of a=6.
	when we perform any arithmetic operation on pointer variable it moves to the next memory address. It increments in their address. Like int a=5,*p;
	p=&a;
	p++;
	Now it increment in their address.

A float variable can store the value of an integer variable.	A float pointer can store the address of a float variable it cannot store the address of integer variable.

Example
int a;
a=5;
	Example
	int *p,b=5;
	p=&b;

It is also known as pointer variation. It mean the slightly difference between the pointer variable and other variable

Dynamic memory allocation

Dynamic memory means, memory which is allocated at run time in place of compile time. In normal array or variable, the memory is allocated at compile time. It is known as static memory. The memory which is allocated at run time is known as dynamic memory. This memory is accessed using pointer. For dynamic memory we can use various methods, which are stored in the **alloc.h header** file.These function are:-

1. **malloc() function**
2. **calloc () function**
3. **realloc() function**
4. **free() function**

1. **malloc()Function:-** malloc means memory allocation function. This function is used to allocate dynamic memory at run time to the pointer variable. This function allocates a single block of memory of specific data type. As shown in the example below:

Write a program to print the elements of pointer using malloc function

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

```
#include<alloc.h>
void main()
{
int i, n,*p;
clrscr();
printf("enter the size of the values");
scanf("%d",&n);
p=(int*) malloc(n);
for(i=0;i<n;i++)
{
printf("enter any number");
scanf("%d",(p+i));
}
for(i=0;i<n;i++)
{
printf("%d\n",(*p));
p++;
}
getch();
}
```

2. **Calloc() Function**:- This function is also used for dynamic memory allocation same as malloc function but this function contain two arguments. The first argument tells the number of elements(items) and the second argument tells their size. It allocates number of blocks memory. As shown in the syntax and example program given below:

void calloc(number of items, size);

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h
> void main()
{
char *p=NULL;
clrscr();
p=(char*) calloc(10,sizeof(char));
printf("enter any string");
scanf("%s",p);
printf("%s",p);
getch();
}
```

3. **Realloc() Function**:- Realloc function means reallocation of memory. This function is used to reallocate memory to the preallocated pointer variable. This function is used to increase and decrease the size of the memory. As shown in the syntax and example below:

```
#include <stdio.h>
#include <alloc.h>
#include <string.h>
#include<conio.h
> void main()
{
  char *p;
  clrscr();
  p = (char *) malloc(10);
  strcpy (p, "CJSOFTECH");
  printf("String is %s”,p);
  printf(“\n Address is %u”, p);
  p = (char *) realloc(p, 20);
  strcpy(p,"CJSOFTECH Patiala");
  printf(“String is %s”,p);
  printf(“\n New address is %u",p);
  getch();
}
```

4. **Free function:-** This function is used to release the memory which is occupied by the variable. That memory, which is allocated by the above discussed functions like malloc(), calloc(), realloc(), so this function is used to free the memory. This function contains a single argument, whose memory we want to release. As shown in the syntax and example below:

void free (pointer variable);
Example
free(*p);

Important Short Answer Type Questions

1. List the applications of pointers in C.
2. What is indirection operator?
3. What is address of operator?
4. What are the advantages of pointers?
5. What is the significance of calloc function in C?
6. What is dynamic memory allocation?
7. What do you mean by pointer to pointer?
8. Why pointer is called double edge weapon? Explain.

Important Long Answer Type Questions

1. Explain with examples the pointer concept in C. State the

advantages of pointers in C.

2. What is a pointer? Explain the difference between simple variable and pointer variable.
3. Demonstrate how pointer can be used to access data of a structure.
4. Describe how pointers are used for handling character string.
5. Write C program-using pointers to find the greatest of given three numbers.
6. Write C program-using pointers to sort the given list of number in ascending order.
7. What do you mean by dynamic memory allocation. Explain different methods used for dynamic memory allocation.