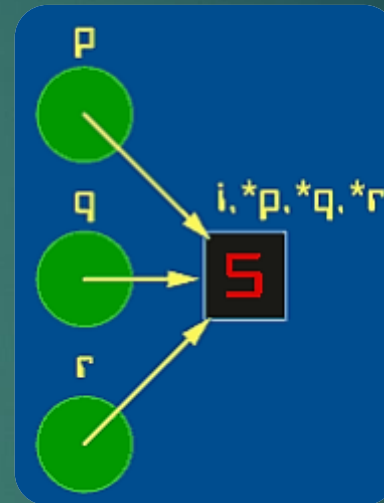


Pointers

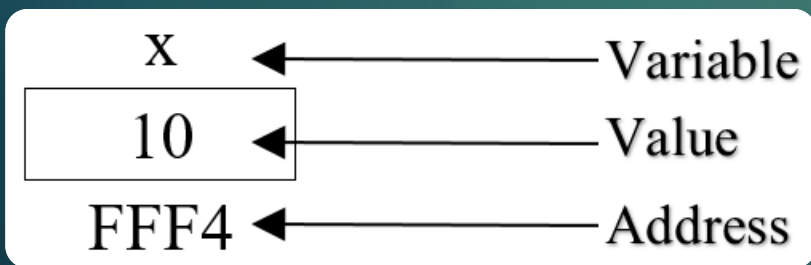
Er. Shiva K. Shrestha (HoD)

Department of Computer Engineering,
Khwopa College of Engineering



Pointer

- ▶ The Concept of Variable Representation
- ▶ Pointer Variable
- ▶ **Reference** or Address Operator (&)
 - ▶ `marks_pointer = &marks;`
- ▶ **Dereference** or Indirection Operator (*)
 - ▶ `*marks_pointer = *(&marks) = marks;`
- ▶ Declaring Variables of Pointer Types
 - ▶ `Data_type * pointer_variable_name;`



j	k
FFF2	5
FFF4	FFF2

Address	Memory Cell	Hex Address
0		0000
1		0001
⋮	⋮	⋮
65514		
65515	FFFC	FFEC
65516		
65517	FFF0	FFEE
65518		
65519	101	FFF0
65520		
65521	FFF4	FFF2
65522		
65523	120	FFF4
65524		
⋮	⋮	⋮
65534		FFFE
65535		FFFF

1. A program to display the contents of the pointer

```
PTR_1.c x
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int x;
5      int *ptr;
6      x=10;
7      clrscr();
8      ptr=&x;
9      printf("x=%d  and ptr=%X",x,ptr);
10     printf("\nx=%d and contents of ptr=%d",x,*ptr);
11     getch();
12 }
```

DOSBox 0.74, Cpu speed: max

x=10 and ptr=FFF4
x=10 and contents of ptr=10

X
10
FFF4

2. Assign a character variable to the pointer & display the contents of the pointer

```
1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     char x,y;
5     char *ptr;
6     x='c'; /*assignment of character*/
7     clrscr();
8     ptr=&x;
9     y=*ptr;
10    printf("Value of x=%c and &x=%X", x,ptr);
11    printf("\nPointer value ptr=%c and &y=%X",y,&y);
12    getch();
13 }
```

DOSBox 0.74, Cpu speed: max 100%

Value of x=c and &x=FFF5
Pointer value ptr=c and &y=FFF4

x	y
c	c
FFF5	FFF4

3. A program to assign the pointer variable to another pointer and display the contents of the both the pointer variables

```

PTR_3.c
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int x;
5      int *ptr1,*ptr2;
6      clrscr();
7      x=10;
8      ptr1=&x;
9      ptr2=ptr1;
10     printf("Value of x=%d and Address of x=%X",x,&x);
11     printf("\nptr1: Content=%d, Value=%X and Address=%X",*ptr1,ptr1,&ptr1);
12     printf("\nptr2: Content=%d, Value=%X and Address=%X",*ptr2,ptr2,&ptr2);
13     getch();
14 }

```

x	ptr1	ptr2
10	FFF4	FFF4
FFF4	FFF2	FFF0



DOSBox 0.74, Cpu speed: max 100% cycles, Frame

```

Value of x=10 and Address of x=FFF4
ptr1: Content=10, Value=FFF4 and Address=FFF2
ptr2: Content=10, Value=FFF4 and Address=FFF0

```

4. A program for assignment and usage of & operator

Er. Shiva K. Shrestha (HoD, Computer Department)
2/5/2019

```
PTR_4_1.c x
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int i=3;
5      clrscr();
6      printf("The address of i is %X",&i);
7      printf("\nValue of i=%d",i);
8      printf("\nValue of i using *(&i) =%d",*(&i));
9      getch();
10 }
```

i
3
FFF4

```
DOS
BOX  DOSBox 0.74, Cpu speed: max
The address of i is FFF4
Value of i=3
Value of i using *(&i) =3_
```

Contd ...

7

```
PTR_4_2.C
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int *j,k=5;
5      clrscr();
6      j=&k;
7      printf("Address of k=%X",&k);
8      printf("\nAddress of j=%X",&j);
9      printf("\nValue of j=%X",j);
10     printf("\nValue of k=%d",k);
11     printf("\nValue of k=%d",*j);
12     getch();
13 }
```

j	k
FFF2	5
FFF4	FFF2

DOSBox 0.74, C

```
Address of k=FFF2
Address of j=FFF4
Value of j=FFF2
Value of k=5
Value of k=5
```


Double Pointer (Pointer to Pointer)

8

Er. Shiva K.
2/5/2019

p	q	r
10	FFF4	<i>FFF2</i>
FFF4	<i>FFF2</i>	FFF0

```

1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int p=10,*q,**r;
5      clrscr();
6      q=&p;
7      r=&q;
8      printf("Address of p=%X",&p);
9      printf("\nAddress of p=%X",q);
10     printf("\nAddress of p=%X",*r);
11     printf("\nAddress of q=%X",&q);
12     printf("\nAddress of q=%X",r);
13     printf("\nAddress of r=%X",&r);
14     printf("\nValue of q=%X",q);
15     printf("\nValue of r=%X",r);
16     printf("\nValue of p=%d",p);
17     printf("\nValue of p=%d",*(&p));
18     printf("\nValue of p=%d",*q);
19     printf("\nValue of p=%d",**r);
20     getch();
21 }
```

DOS
BOX

DOSBox 0.74, Cpu

```

Address of p=FFF4
Address of p=FFF4
Address of p=FFF4
Address of q=FFF2
Address of q=FFF2
Address of r=FFF0
Value of q=FFF4
Value of r=FFF2
Value of p=10
Value of p=10
Value of p=10
Value of p=10_
```


5. A program to increment the pointer's address

9

```
PTR_5.c x
1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     int value,*ptr;
5     clrscr();
6     value=120;
7     clrscr();
8     ptr=&value;
9     printf("ptr: Content=%d, Value=%X, Address=%X",*ptr,ptr,&ptr);
10    ptr++;
11    printf("\nptr: Content=%X, Value=%X, Address=%X",*ptr,ptr,&ptr);
12    getch();
13 }
```



DOSBox 0.74, Cpu speed: max 100% cycles, Fra

```
ptr: Content=120, Value=FFF4, Address=FFF2
ptr: Content=0, Value=FFF6, Address=FFF2_
```

value	ptr	x	ptr1	ptr2
120	FFF4	101	FFF0	FFFC
FFF4	FFF2	FFF0	FFEE	FFEC
65524	65522	65520	65518	65516

```

1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     int value,*p;
5     int x,*p1,*p2;
6     clrscr();
7     printf("&value=%X, &p=%X, &x=%X, &p1=%X, &p2=%X",&value,&p,&x,&p1,&p2);
8     printf("\n&value=%u, &p=%u, &x=%u, &p1=%u, &p2=%u\n",&value,&p,&x,&p1,&p2);
9     value=120;
10    p=&value;
11    printf("\nValue of p=%X",p);
12    p++;
13    printf("\nMemory address after increment=%X",p);
14    p--;
15    printf("\nMemory address after decrement=%X",p);
16    x=101;
17    printf("\nx: Value=%d, Address=%X",x,&x);
18    p1=&x;
19    printf("\np1: Content=%d, Value=%X, Address=%X",*p1,p1,&p1);
20    p2=p1+2; /* 2*2 Bytes */
21    printf("\nAddress of (p2=p1+2)=%X",&p2);
22    printf("\np2: Content=%X, Value=%X, Address=%X",*p2,p2,&p2);
23    getch();
24 }

```

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Pro

&value=FFF4, &p=FFF2, &x=FFF0, &p1=FFEE, &p2=FFEC
&value=65524, &p=65522, &x=65520, &p1=65518, &p2=65516

Value of p=FFF4
Memory address after increment=FFF6
Memory address after decrement=FFF4
x: Value=101, Address=FFF0
p1: Content=101, Value=FFF0, Address=FFEE
Address of (p2=p1+2)=FFEC
p2: Content=78, Value=FFF4, Address=FFEC_

Address	Memory Cell	Hex Address
0		0000
1		0001
⋮	⋮	⋮
65514		
65515	FFFC	FFEC
65516		
65517	FFF0	FFEE
65518		
65519	101	FFF0
65520		
65521	FFF4	FFF2
65522		
65523	120	FFF4
65524		
⋮	⋮	⋮
65534		FFFE
65535		FFFF

Some Terminologies

- ▶ Bad Pointer – Declared, but uninitialized pointer
- ▶ Void Pointer – points to variables of any data type
- ▶ Null Pointer – points nowhere or nothing
- ▶ Double Pointer – `**ptr`
- ▶ Array of Pointer – `data_type *pointer_name[size]`

Pointers & Arrays

Equivalent Expressions of Arrays & Pointers

Type of Array	To be Accessed	Technique	
		Array Notation	Pointer Notation
1D	Address of i th Element	<code>&marks[i]</code>	<code>(marks+i)</code>
	Value of i th Element	<code>marks[i]</code>	<code>*(marks+i)</code>
2D	Address of Element at i th Row & j th Column	<code>&marks[i][j]</code>	<code>*(marks+i)+j</code>
	Value of Element at i th Row & j th Column	<code>marks[i][j]</code>	<code>*(*(marks+i)+j)</code>

Remember: $i = *(&i)$

Relationship between 1D Array & Pointer

Q. WAP to access the array element using pointer

13

Note: $i = *(&i)$

```
DOS BOX  DOSBox 0.74, Cpu speed: max 100% cycles, Framerate: 100%
1         2         3         4         5         _
```

```
PTR_12_1.c  PTR_12_2.c  PTR_12_3.C
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int myArray[]={1,2,3,4,5};
5      int *ptr2myArray,i;
6      clrscr();
7      ptr2myArray=myArray;
8      for(i=0;i<5;i++){
9          printf("%d ",*(ptr2myArray++));
10     }
11     getch();
12 }
```

```
PTR_12.c  PTR_12_1.c  PTR_12_2.c
1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int i,myArray[5]={1,2,3,4,5};
5      clrscr();
6      for(i=0;i<5;i++){
7          printf("%d\t",*(myArray+i));
8          //printf("%d\t",myArray[i]);
9      }
10     getch();
11 }
```

```
DOS BOX  DOSBox 0.74
1 2 3 4 5
```

Array is internal pointer ...

String & Pointer Array - Internal Pointer

PTR_13_v2.c

```
1 #include<conio.h>
2 #include<stdio.h>
3 int main(){
4     char x[]="Welcome to Khwopa!\n";
5     char *y="C Programming is easy.";
6     printf("%s",x);
7     printf("%s",y);
8     getch();
9     return 0;
10 }
```

```
int x[5]={1,2,3,4,5};
int *p;
p=x; /* p=&x[0] */
```

Element	x[0]	x[1]	x[2]	x[3]	x[4]
Value	1	2	3	4	5
Address	1000	1002	1004	1006	1008

```
p=&x[0];
p=&x[1];
p=&x[2];
p=&x[3];
p=&x[4];
```

HOD, Computer Department)

C:\Users\ErSKS\Google Drive (c.khwopa

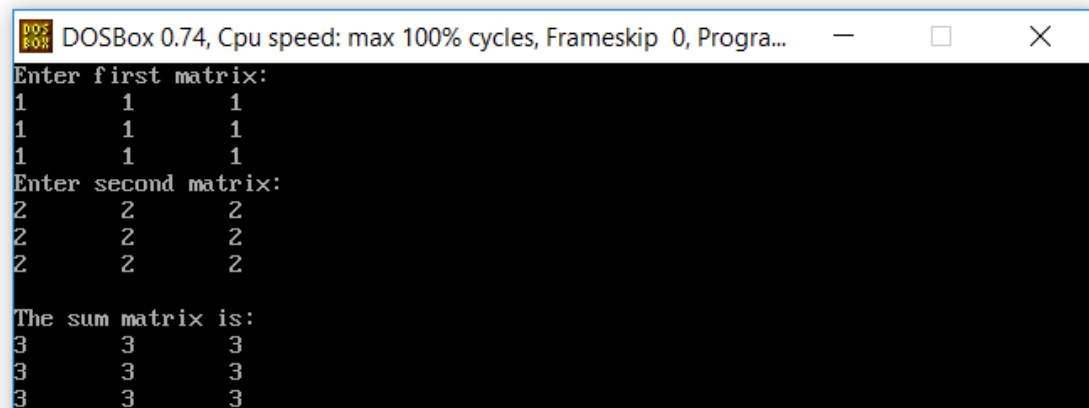
```
Welcome to Khwopa!
C Programming is easy._
```


2D Array & Pointer

15

WAP to add two 3*3 matrices using pointer

```
P17_V2.C
1 #include<stdio.h>
2 #include<conio.h>
3 #define M 3
4 #define N 3
5 int main(){
6     int i, j;
7     int (*a)[N], (*b)[N], (*sum)[N];
8     clrscr();
9     printf("Enter first matrix:\n");
10    for (i = 0; i < M; i++){
11        for (j = 0; j < N; j++){
12            //scanf("%d",&a[i][j]);
13            scanf("%d",*(a+i)+j);
14        }
15    }
16    printf("Enter second matrix:\n");
17    for (i = 0; i < M; i++){
18        for (j = 0; j < N; j++){
19            //scanf("%d",&b[i][j]);
20            scanf("%d",*(b+i)+j);
21        }
22    }
23    printf("\nThe sum matrix is:\n");
24    for (i = 0; i < M; i++){
25        for (j = 0; j < N; j++){
26            //sum[i][j] = a[i][j] + b[i][j]
27            (*(sum+i)+j) = (*(a+i)+j) + (*(b+i)+j);
28            //printf("%d\t", sum[i][j]);
29            printf("%d\t", (*(sum+i)+j));
30        }
31        printf("\n");
32    }
33    getch();
34    return 0;
35 }
```



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter first matrix:
1 1 1
1 1 1
1 1 1
Enter second matrix:
2 2 2
2 2 2
2 2 2
The sum matrix is:
3 3 3
3 3 3
3 3 3
```

Pointer Arithmetic

16

C support four arithmetic operators

1. Addition +
2. Subtraction -

3. Increment ++
4. Decrement --

```
1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     int x,y,*ptr;
5     clrscr();
6     x=10;
7     ptr=&x;
8     printf("Value of x=%d and pointer=%d",x,*ptr);
9     y=++ *ptr; /*y=11*/
10    printf("\nValue of y=%d and pointer=%d",y,*ptr);
11    getch();
12 }
```

x	y	ptr
10	11	&x
&x	&y	&ptr

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
Value of x=10 and pointer=10
Value of y=11 and pointer=11_
```

```

1  #include<stdio.h>
2  #include<conio.h>
3  void main(){
4      int x,y,*x_pointer,temp;
5      clrscr();
6      temp=3;
7      x=5*(temp+5);|
8      x_pointer=&temp;
9      y=6*(*x_pointer+5);
10     printf("x=%d",x);
11     printf("\ny=%d",y);
12     printf("\nAddress of temp=%X",&temp);
13     getch();
14 }

```



DOSBox 0.74, Cpu sp

x=40

y=48

Address of temp=FFF0

x	y	x_pointer	temp
40	48	FFF0	3

FFF0

```

1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     int x,y,*p1,*p2;
5     clrscr();
6     x=25;
7     p1=&x;
8     printf("&x=%X, &y=%X, &p1=%X, &p2=%X\n",&x,&y,&p1,&p2);
9     printf("Contents of pointer=%d",*p1);
10    *p1=*p1+1; // ++*p1
11    y=*p1;
12    printf("\nx=%d, y=%d and (*p1=*p1+1)=%d",x,y,*p1);
13    (*p1)++;
14    y=*p1;
15    printf("\nx=%d, y=%d and (*p1)++=%d",x,y,*p1);
16    ++ *p1;
17    y=*p1;
18    printf("\nx=%d, y=%d and ++ *p1=%d",x,y,++ *p1);
19    ++ *p1;
20    p2=p1;
21    printf("\n\n*p1=%d",*p1);
22    printf("\n*p2=%d",*p2);
23    getch();
24 }

```

x	y	ptr	ptr2
25	26	&x	&x
&x	&y	&ptr	&ptr2
26	28		
27			
28			
29			
30			

18

Er. Shiva K. Shrestha (HoD, Computer Department)
2/5/2019

DOS
BOX

DOSBox 0.74, Cpu speed: max 100% cycles.

```

&x=FFF4, &y=FFF2, &ptr=FFF0, &ptr2=FFEE
Contents of pointer=25
x=26, y=26 and (*ptr=*ptr+1)=26
x=27, y=27 and (*ptr)++=27
x=29, y=28 and ++ *ptr=29

*ptr=30
*ptr2=30_

```

Pointer Operations

19

Er. Shiva K. Shrestha (HoD, Computer Department)
2/5/2019

Valid Operations

- ▶ Assignment to a pointer of the same type
- ▶ Assigning a pointer to pointer of type (void *) and back
- ▶ Adding/subtracting a pointer & an integer (including increment & decrement)
- ▶ Subtracting or comparing two pointers which point to members of the same array
- ▶ Assigning or comparing to zero

Invalid Operations

- ▶ Adding two pointers
- ▶ Multiply, divide, shift, mask pointers
- ▶ Add float or double numbers to a pointer
- ▶ Assign a pointer of different types without cast

Valid Pointer Operations

20

Er. Shiva K. Shrestha (HoD, Computer Department)
2/5/2019

```
1  #include<stdio.h>
2  #include<conio.h>
3  #define NULL 0
4  void main(){
5      int x,y;
6      int *px=&x;
7      int *py;
8      void *pv;
9      clrscr();
10     py=px;           /* Assignment to ptr of same type*/
11     px=(int *) pv;   /* recast a (void *) pointer */
12     pv=(void *)px;   /* recast to type (void *) */
13     py=px+2;         /* Add/Sub ptr & integer is legal */
14     px++;            /* Inc/Dec is legal*/
15     if(px==NULL)     /* Compare to null pointer*/
16     py=NULL;         /* Assign to null pointer*/
17     getch();
18 }
```


Invalid Pointer Operations

21

```
PTR_9.c x
1 #include<stdio.h>
2 #include<conio.h>
3 #define NULL 0
4 void main(){
5     int x,y;
6     int *px,*py,*p;
7     float *pf;
8     clrscr();
9     px=&x;
10    py=&y;
11    p=px+py;
12    p=px*py;
13    p=px/py;
14    p=px+10.5;
15    pf=px;
16    getch();
17 }
```

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
File Edit Search Run Compile Debug Project Options Window Help
[ ] Message 2=[+]-
Compiling PTR_9.C:
•Error PTR_9.C 11: Invalid pointer addition
Error PTR_9.C 12: Illegal use of pointer
Error PTR_9.C 13: Illegal use of pointer
Error PTR_9.C 14: Illegal use of floating point
Warning PTR_9.C 15: Suspicious pointer conversion
Warning PTR_9.C 17: 'pf' is assigned a value that is never used
Warning PTR_9.C 17: 'py' is assigned a value that is never used
```

```
/*Addition of two pointer is illegal */
/* Multiplication of two pointer is illegal */
/* Division of two pointer is illegal */
/* Addition of float to pointer is illegal */
/* Assignment of different types of pointer is illegal */
```

Pointers and Functions

22

Call By Reference

```
PTR_10.C
1 #include<stdio.h>
2 #include<conio.h>
3 void swap(int a,int b){
4     int temp;
5     temp=a;
6     a=b;
7     b=temp;
8 }
9 void main(){
10     int x=100,y=20;
11     clrscr();
12     printf("Values before swap\n");
13     printf("x=%d and y=%d",x,y);
14     swap(x,y);
15     printf("\nValues after swap\n");
16     printf("x=%d and y=%d",x,y);
17     getch();
18 }
```

```
DOSBox 0.74, Cpu speed:
Values before swap
x=100 and y=20
Values after swap
x=100 and y=20_
```

```
PTR_11.C
1 #include<stdio.h>
2 #include<conio.h>
3 void swap(int *a,int *b){
4     int temp;
5     temp=*a;
6     *a=*b;
7     *b=temp;
8 }
9 void main(){
10     int x=100,y=20;
11     printf("Values before swap\n");
12     printf("x=%d and y=%d",x,y);
13     swap(&x,&y);
14     printf("\nValues after swap\n");
15     printf("x=%d and y=%d",x,y);
16 }
```

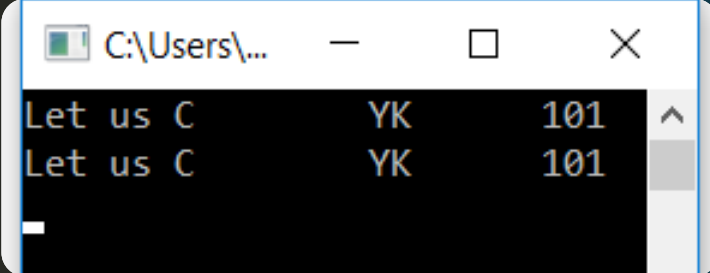
```
DOSBox 0.74, Cpu speed:
Values before swap
x=100 and y=20
Values after swap
x=20 and y=100
```

Structure & Pointer

C language allows declaring pointer to structure just like pointer to other ordinary variables. The declaration can be done in the following manner:

```
struct employee{  
    char *name;  
    char *roll_no;  
    int salary;  
}*emp1;
```

```
1  #include<stdio.h>  
2  #include<conio.h>  
3  int main(){  
4      struct book{  
5          char name[25];  
6          char author[25];  
7          int call_no;  
8      };  
9      struct book b1={"Let us C","YK",101};  
10     struct book *ptr;  
11     ptr=&b1;  
12     printf("%s\t%s\t%d\n",b1.name,b1.author,b1.call_no);  
13     printf("%s\t%s\t%d\n",ptr->name, ptr->author, ptr->call_no);  
14     getch();  
15     return 0;  
16 }
```



Let us C	YK	101
Let us C	YK	101

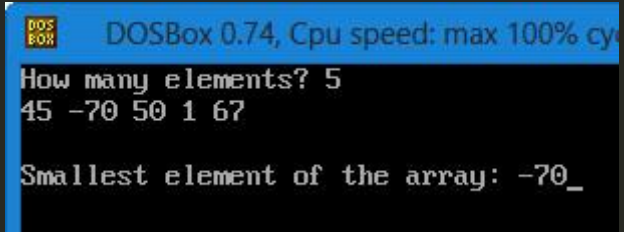
Applications of Pointer

- ▶ enhances execution speed of a program
- ▶ saves memory space
- ▶ used to construct different data structures such as linked lists, queues, stacks, etc.
- ▶ supports dynamic allocations and de-allocations of memory segments

Find smallest no. using pointer

25

```
PTR_15.C
1 #include<stdio.h>
2 #include<conio.h>
3 void main(){
4     int i,n,small,*ptr,a[10];
5     clrscr();
6     printf("How many elements? ");
7     scanf("%d",&n);
8     for(i=0;i<n;i++){
9         scanf("%d",&a[i]);
10    }
11    //Assign address of a[0] to pointer variable
12    //It can be done in two ways ptr=&a[0] or ptr=a
13    ptr=a;
14    small=*ptr;
15    ptr++; //Pointer points to next location in an array
16    for(i=1;i<n;i++){ //loop n-1 times to search smallest element
17        if(small>*ptr){
18            small=*ptr;
19        }
20        ptr++; //pointer is incremented to pointed a[i+1]
21    }
22    printf("\nSmallest element of the array: %d",small);
23    getch();
24 }
```



DOSBox 0.74, Cpu speed: max 100% cy

How many elements? 5
45 -70 50 1 67

Smallest element of the array: -70_

Re: Pointer & Function

26

Er. Shiva K. Shrestha (HOD, Computer Department)
2/5/2019

```
1  #include<stdio.h>
2  #include<conio.h>
3  int *larger(int *,int *); // Prototype
4  void main(){
5      int a=10,b=20,*p;
6      clrscr();
7      p=larger(&a,&b); // Function Call
8      printf("Largest Value=%d",*p);
9      getch();
10 }
11 int *larger(int *x,int *y){
12     if (*x>*y){
13         return(x); // Address of a
14     }else{
15         return(y); // Address of b
16     }
17 }
```

DOSBox 0.74, Cpu speed:
Largest Value=20_

Q/A?

27

Thank You!

Er. Shiva K. Shrestha

computer.khwopa@gmail.com

Er. Shiva K. Shrestha (HoD, Computer Department)
2/5/2019