LOOPS & INTRODUCTION TO POINTERS

BY:

DR. SHEFALI ARORA CHOUHAN

ASSISTANT PROFESSOR CSE

NIT JALANDHAR

LOOP EXAMPLES:
FIND SUM OF NUMBERS

```c
#include <stdio.h>

int main()
{
    int i,n;
    int sum=0;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        sum+=i;

    }
    printf("%d",sum);
}
```

input

```
4
10
```

FIND SUM OF DIGITS OF A NUMBER

```
 8  ************************************************
 9  #include <stdio.h>
10
11  int main()
12  {
13      int n,rem;
14      int sum=0;
15      printf("Enter no");
16      scanf("%d",&n);
17      while(n>0)
18      {
19          rem=n%10;
20          sum+=rem;
21          n=n/10;
22      }
23      printf("Sum is %d",sum);
24  }
```

```
Enter no121
Sum is 4

...Program finished with exit code 0
Press ENTER to exit console.
```

PRINT MULTIPLICATION TABLE FROM 2 TO 4

```c
/*****************************************************************
#include <stdio.h>

int main()
{
    int i;
    for(i=2;i<=4;i++)
    {
    for(int j=1;j<=10;j++)
    {
        printf("%d * %d = %d \n",i,j,i*j);
    }
    }
    return 0;
}
```

input

```
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
```

BREAK VS CONTINUE

```c
#include <stdio.h>

int main()
{
    int i;
    for(i=2;i<=4;i++)
    {
      printf("%d \n",i);
      if(i==3)
        break;
    }
}
```

```
2
3
```

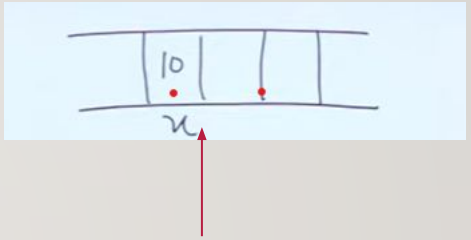
```c
C#, OCaml, VB, Perl, Swift, Prolog, Javas
Code, Compile, Run and Debug online from

*****************************************
#include <stdio.h>

int main()
{
    int i;
    for(i=2;i<=4;i++)
    {
      printf("%d \n",i);
      if(i==3)
        continue;
    }
}
```

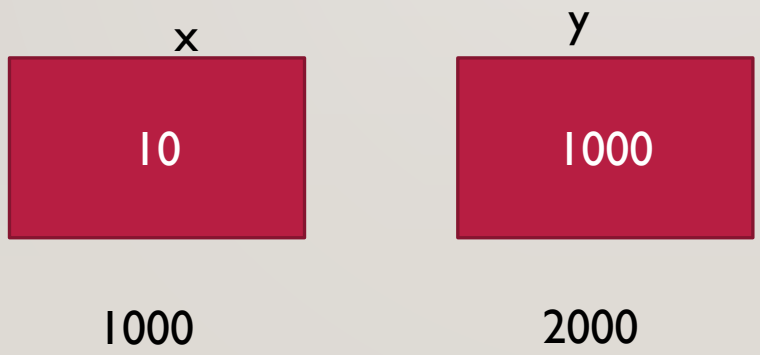
```
3
4
```

POINTERS

USE OF POINTERS

- Stores addresses of variables
- * is used for derefencing
- & is used for referencing
- We store address of a variable in a pointer





POINTER ARITHMETIC : ADDITION

main.c

```
4  GDB online is an online compiler and debugger tool for C, C++
5  C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML,
6  Code, Compile, Run and Debug online from anywhere in world.
7
8  *******************************************************************
9  #include <stdio.h>
10
11 int main()
12 {
13     unsigned int x=10;
14     int *y=&x;
15     printf("%p %d \n",y,*y);
16     y=y+3;
17     printf("%p %d",y,*y);
18
19 }
```

input

```
0x7fff96b56ebc 10
0x7fff96b56ec8 1454999296
```

- Pointer moves three positions forward
- Will print value at the given address
- Use of – will take it three positions back

POINTER INCREMENT / DECREMENT

main.c

```c
int main()
{
    int x=10;
    int *y=&x;
    int b=9;
    int *z=&b;
    printf("%u %d \n", y,*y);
    printf("%u %d \n", z,*z);
    y++;
    z++;
    printf("%u %d \n", y,*y);
    printf("%u %d \n", z,*z);
    y--;
    z--;
    printf("%u %d \n", y,*y);
    printf("%u %d \n", z,*z);


}
```

input

```
1209398832 10
1209398836 9
1209398836 9
1209398840 1209398836
1209398832 10
1209398836 9
```

VOID POINTER

- Void pointer has no associated data type
- It can be typecasted to any type
- Int main()
- {
- Int n=10;
- void *x= &n;
- printf("%d",*(int*)x);
- }

SIZEOF() ON POINTERS

main.c

```
Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************************
#include <stdio.h>

int main()
{
    unsigned int x=10;
    int *y=&x;
    float b=7.5;
    float *z=&b;
    printf("%p %d %d \n",y,*y, sizeof(y));
    printf("%p %f %d \n",z,*z, sizeof(z));

}
```

input

```
      |                   int          long unsigned int
      |                  %ld
0x7fffcae4d5d0 10 8
0x7fffcae4d5d4 7.500000 8
```

SUBTRACTION OF POINTERS

- Gives the offset between addresses of variables

main.c

```c
/*******************************************************************************

Welcome to GDB Online.
GDB online is an online compiler and debugger tool for C, C++, Python, PH
C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, HTML, CSS, JS
Code, Compile, Run and Debug online from anywhere in world.

*******************************************************************************/
#include <stdio.h>

int main()
{
    int x=10;
    int *a=&x;
   int y=15;
   int *b=&y;
   printf("%u %u %ld \n",a, b, b-a);
   printf("%d",*b-*a);

}

```

input

```
   17 |    printf("%u %u %ld \n",a, b, b-a);
      |                  ~^             ~
      |                   |             |
      |                   unsigned int int *
      |                  %ls
4036614336 4036614340 1
5
```

GIVE OUTPUT?

- int x[]={10,20};
- int *p=&x;
- *p++; // 20 ; ++ gets executed first;
- ++*p; //11 as it increments value of *p i.e 10 Same as ++(*p)
- *++p // 20

```c
/*****************************************************************/
#include <stdio.h>

int main()
{
    int x=10;
    int *y=&x;
    printf("%d \n %d \n %d \n%d \n %d",x,y, &x, &y,*y);
}
```

input

```
10
 -2004362340
 -2004362340
-2004362336
 10
```

ILLEGAL OPERATIONS USING POINTERS

- Address + Address = illegal
- Address * Address = illegal
- Address % Address = illegal
- Address / Address = illegal
- Address & Address = illegal
- Address ^ Address = illegal
- Address | Address = illegal
- ~Address = illegal