SMR3821: C and Linux for Science Engineering:

Introduction to C Programming

Pointers

Elliot Menkah

elliotsmenkah@gmail.com

November 24, 2022

National Institute for Mathematical Sciences Ghana

Outline

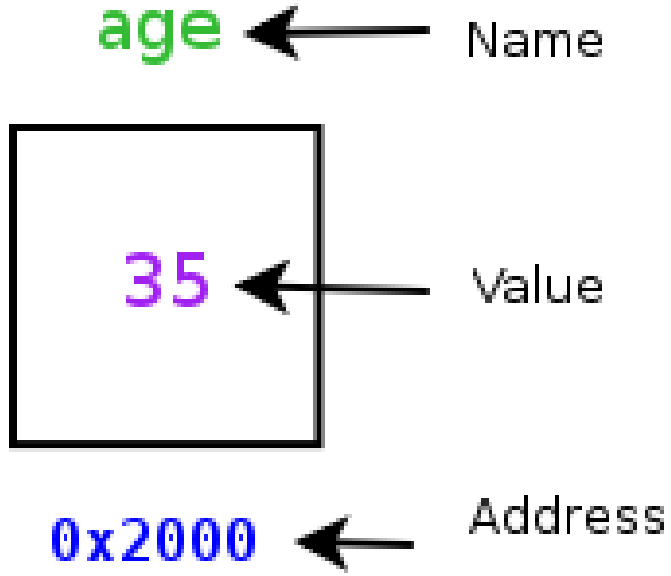
where are we?

National Institute for Mathematical Sciences Ghana

Variables Revisited

Recall that a variable has the following characteristics

1. name
2. address
3. value
4. type



How can we obtain the address of a variable?

By using the 'Address Of' operator **&**

ILLUSTRATION:

```c
#include<stdio.h>

int main()
{
        float X = 302.45;

        printf("Value of X = %f\n",X);
        //Use of Address Of operator &
        printf("Memory Address of X = %p\n",&X);

}
```

Given a memory address, how do we determine the value at that location?

By using the 'Value At Address' Or Indirection operator *

ILLUSTRATION:

```
1 #include<stdio.h>
2
3 int main()
4 {
5         float X = 302.45;
6         //The usual way
7         printf("Value of X = %f\n",X);
8         //The use of the 'Value At Address' operato
9         printf("Value of X = %f\n", *(&X)   );
10
11
12 }
```

National Institute for Mathematical Sciences Ghana

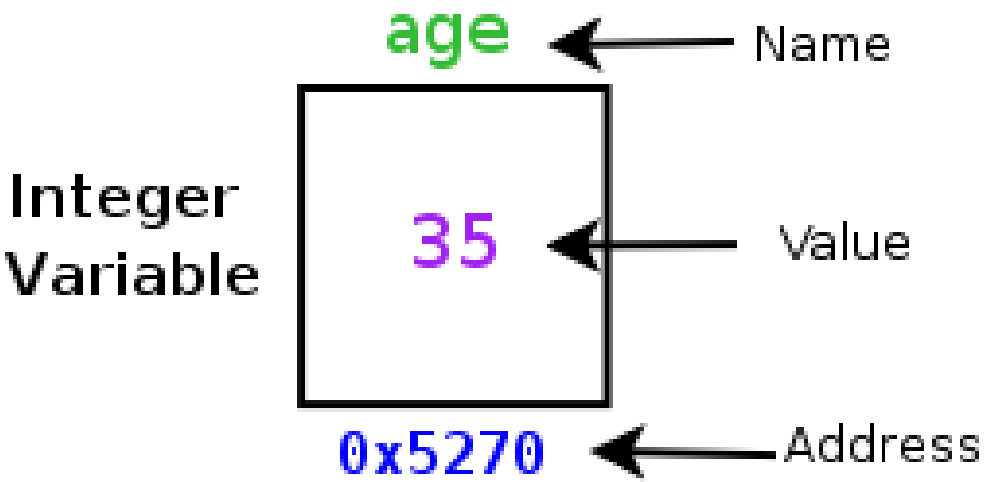
What is a Pointer?

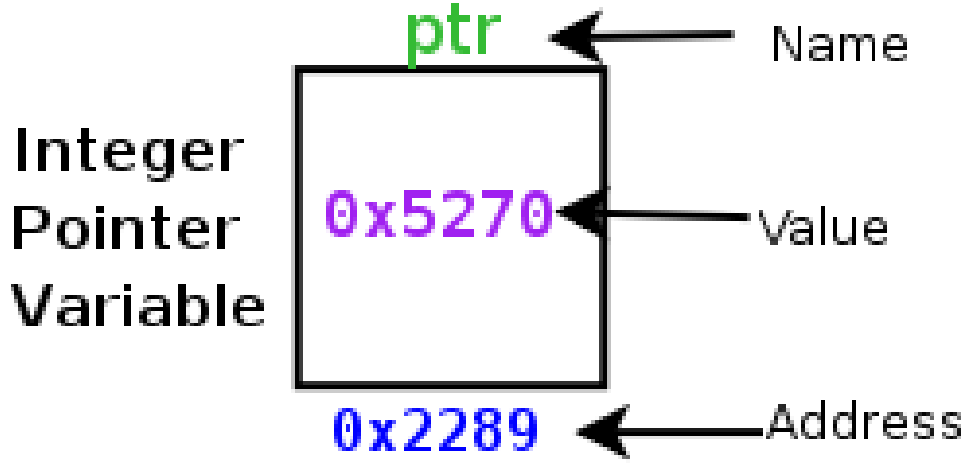
A Pointer is a variable that holds memory address values.

Since a pointer is still a variable, the following characteristics still apply.

1. name
2. address
3. value
4. type

The type a of pointer determines the kind of address it can store. For example, if we say a pointer is of type int, then only the memory address of integer variables can be stored by the pointer.


age
Name
Integer Variable
35
Value
0x5270
Address


ptr
Name
Integer Pointer Variable
0x5270
Value
0x2289
Address

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

Syntax: Declaration of a Pointer

```
data_type * pointer_name;

//declare an integer pointer
int*iPtr;

//declare a character pointer
char * pointer;

//declare a float pointer
float *direct;

//declare a pointer to an integer pointer
int ** k;
```

National Institute for Mathematical Sciences Ghana

Syntax: Example Using Pointers

```c
#include<stdio.h>

int main()
{
        int i = 200;      //integer variable
        int * j;          //integer pointer
        int ** k;         //pointer to an integer pointer

        j=&i;             //assign address of i to j
        k=&j;             //assign address of j to k

        printf("Addr. of i = %p\n",&i);
        printf("Addr. of i = %p\n",j);
        printf("Val. of i = %d\n",i);
        printf("Val. of i = %d\n",*(&i));
        printf("Val. of i = %d\n",*j);
        printf("Val. of i = %d\n",**k);

}
```

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

Passing Arguments By Value

Pass By Value Copies of values of actual arguments are passed on to their corresponding formal arguments during function invocation.

Example

```c
#include<stdio.h>
void swap(int,int);
int main()
{
   int i=5;
   int j=10;
   printf("\nIn main :: Before swap: i = %d, j = %d",i,j);
   swap(i,j);
   printf("\nIn main :: After swap: i = %d, j = %d\n",i,j);
}

void swap(int a, int b)
{
   int tmp;
   printf("\nIn swap :: Before swap: a = %d, b = %d",a,b);
   tmp=a;
   a=b;
   b=tmp;
   printf("\nIn swap :: After swap: a = %d, b = %d",a,b);
}
```

National Institute for Mathematical Sciences Ghana

Passing Arguments By Reference

Pass By Reference Addresses of actual arguments are passed on to their corresponding formal arguments duing function invocation.

Example

```c
#include<stdio.h>
void swap(int*,int*);
int main()
{
   int i=5;
   int j=10;
   printf("\nIn main :: Before swap: i = %d, j = %d",i,j);
   swap(&i,&j);
   printf("\nIn main :: After swap: i = %d, j = %d\n",i,j);
}

void swap(int * a, int * b)
{
   int tmp;
   printf("\nIn swap :: Before swap: a = %d, b = %d",*a,*b);
   tmp=*a;
   *a=*b;
   *b=tmp;
   printf("\nIn swap :: After swap: a = %d, b = %d",*a,*b);
}
```

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

Pointers In Arithmetic Expressions

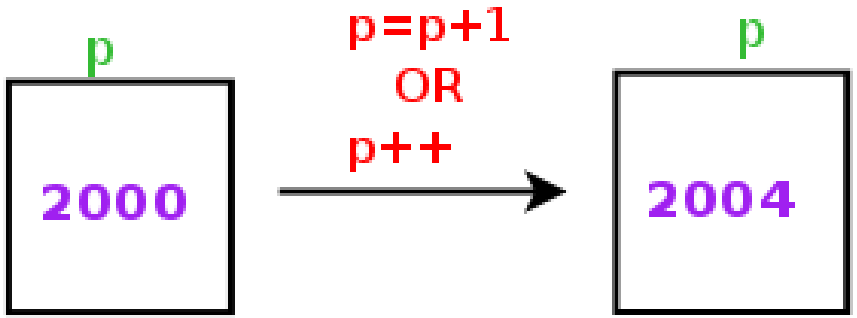
Pointers can be used in arithmetic expression as illustrated below: Example

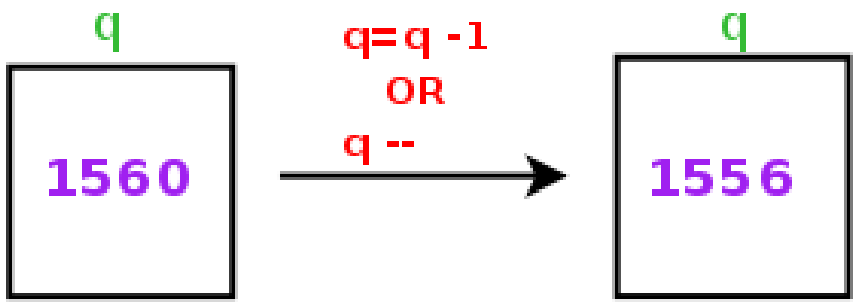
```
1 #include<stdio.h>
2
3 int main()
4 {
5         int x=200;
6         int y=400;
7         int * ptrX;
8         int * ptrY;
9
10        ptrX=&x;
11        ptrY=&y;
12
13        printf("\nx + *ptrX = %d",x + *ptrX);
14        printf("\ny - *ptrY = %d",y - *ptrY);
15        printf("\n5 * *ptrX - *ptrY = %d",5* *ptrX - *ptrY);
16        printf("\n*ptrX * *ptrY = %d",*ptrX * *ptrY);
17        printf("\n*ptrY / *ptrX = %d\n",*ptrX/ *ptrY);
18 }
```

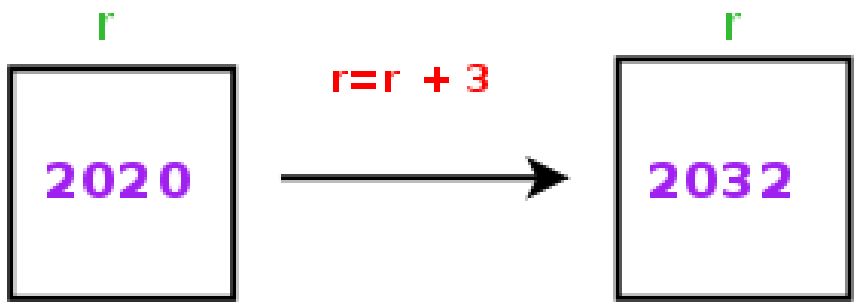
National Institute for Mathematical Sciences Ghana

Pointer Arithmetic

Suppose p,q and r are integer pointers

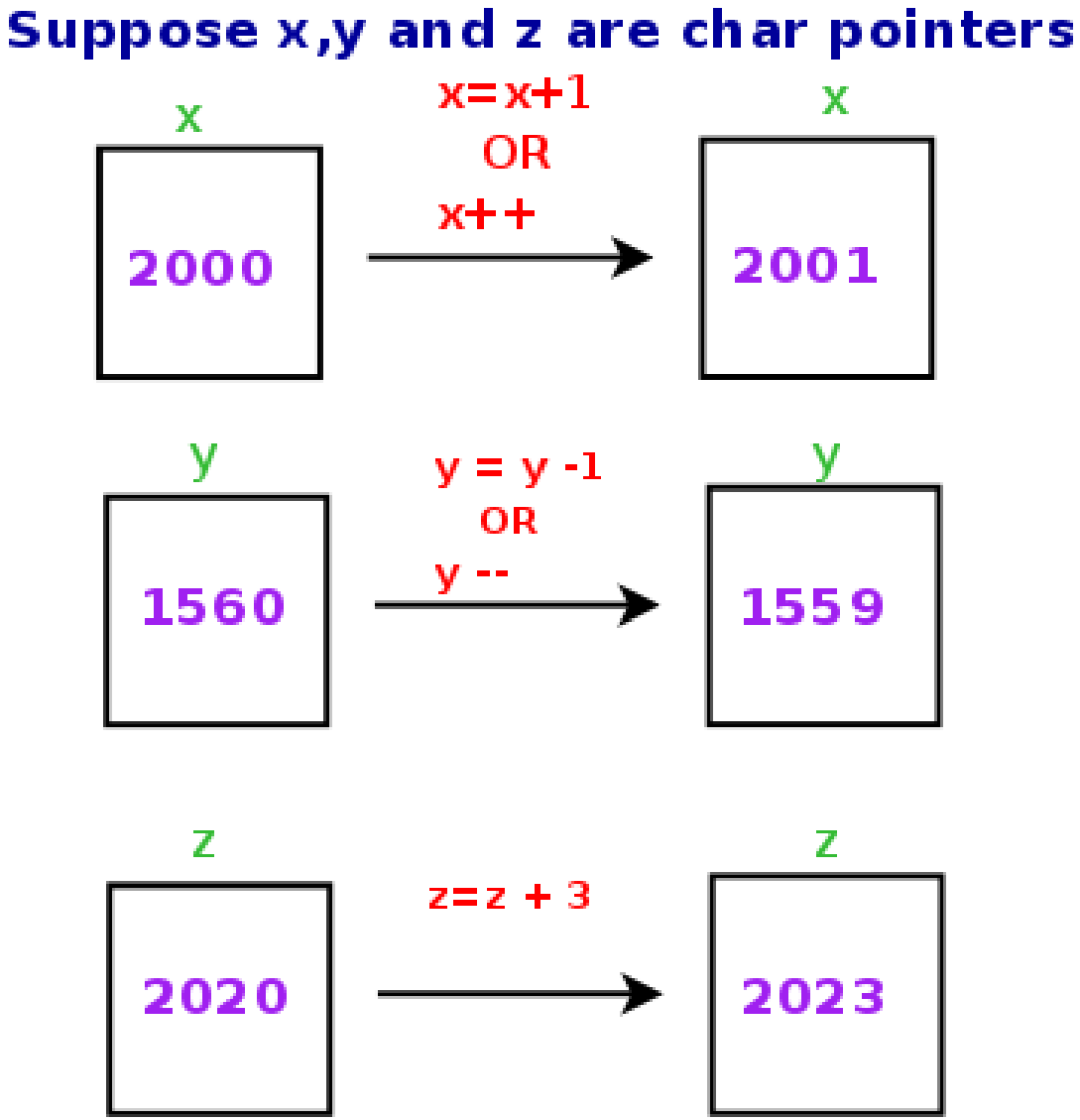






National Institute for Mathematical Sciences Ghana

Pointer Arithmetic



National Institute for Mathematical Sciences Ghana

Pointer Arithmetic

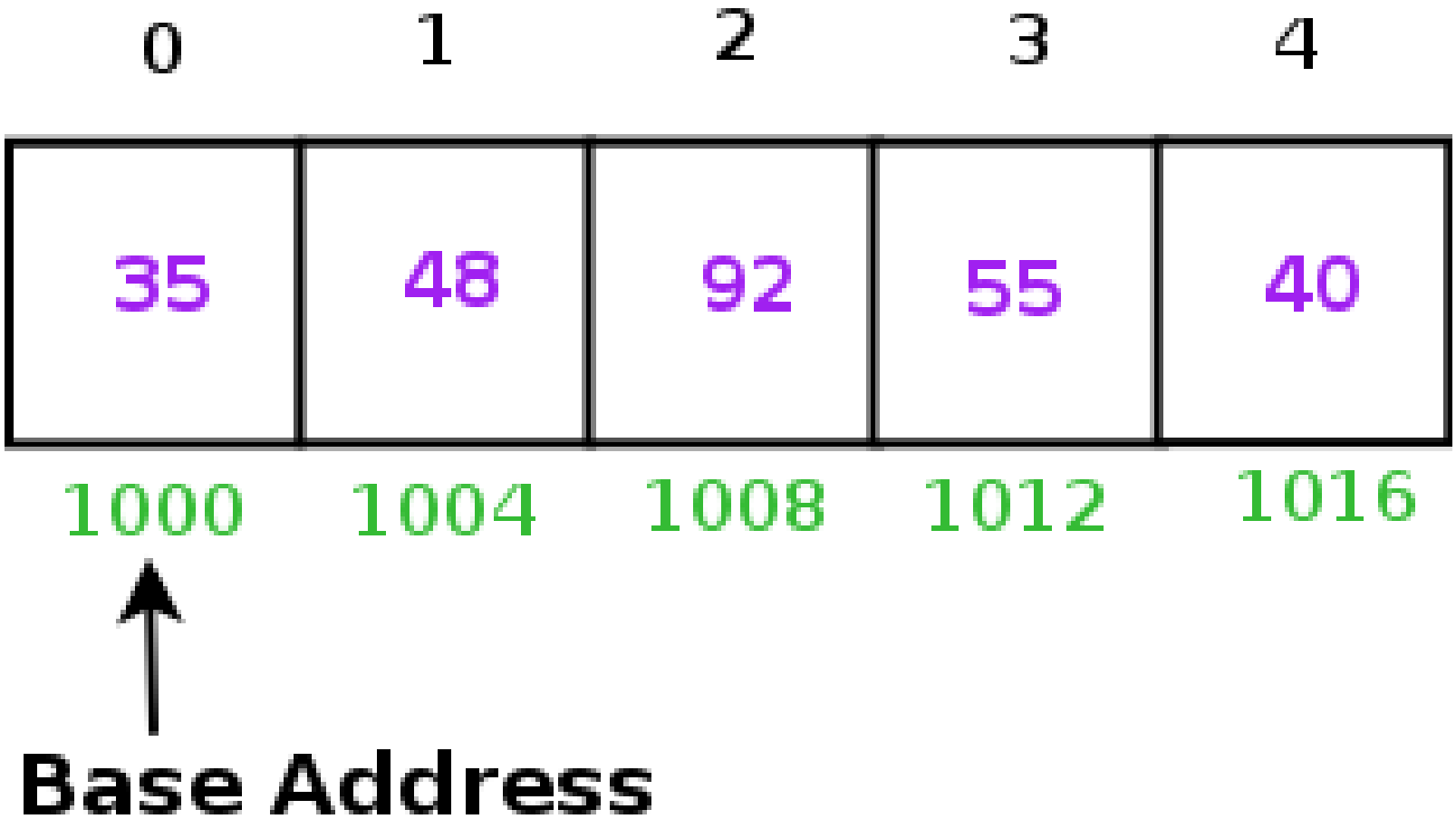
1. When we increment a pointer, its value is increased by the length of the data type it points to.
   - char (1 byte)
   - integer (4 bytes) 32 bit word computer
   - float (4 bytes)
   - double (8 bytes)
2. The length is refered to as the scale factor

National Institute for Mathematical Sciences Ghana

where are we?

National Institute for Mathematical Sciences Ghana

Pointers & Arrays



National Institute for Mathematical Sciences Ghana

Pointers & Arrays

The address of the first element of an array is called the Base Address

The base address of an array may be assigned to a pointer in two way:

```
1 int marks[5];
2 int * ptr;
3
4 //Assign base address of marks to ptr
5 ptr=&marks[0];
6
7 //Alternatively, assign base address of marks to pt
8 ptr=marks;
```

National Institute for Mathematical Sciences Ghana

Example of Pointers and Arrays

```c
#include<stdio.h>

int main()
{
        int marks[5]={30,20,70,25,55};
        int * ptr;
        int i=0;

        //assign base address of marks to ptr
        ptr=marks;

        for(i=0;i<5;i++)
        {
                printf("@ %u: marks[%d] = %d\n",(ptr+i), i,

        }
}
```

National Institute for Mathematical Sciences Ghana

Speed Byte

Question 1 :

With reference to question 1 on arrays, previously discussed, where you find fx values for given x, write a C code that has a void function called swap that uses pointer(s) to swap the values of x and fx and prints the results to screen.
The swap function should take two arguments:

swap(int * a, int * b)

Special thanks to the following for their initial work:

- *Kwesi A. Smith*
- *Shirley A. Akasreku*
- *Ernie Ofori*
- *Peter Amoako Yirenkyi*