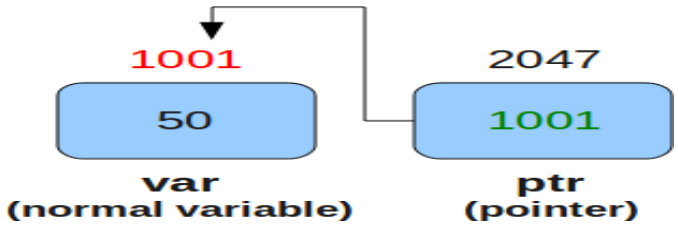
Department of Electrical and Computer Engineering, NSU
CSE 115L: Fundamentals of Computer Programming
Week 09 (Pointers)

Pointers: C Pointer is a variable that stores/points the address of another variable. C Pointer is used to allocate memory dynamically i.e. at run time. The pointer variable might be belonging to any of the data type such as int, float, char, double, short etc.



Example 1: Manipulating address and values with pointer

```
#include <stdio.h>
int main(){

   int c;
   int* pc;
   c=22;
   printf("Address of c:%d\n",&c);
   printf("Value of c:%d\n\n",c);
   pc=&c;
   printf("Address of pointer pc:%d\n",pc);
   printf("Content of pointer pc:%d\n\n",*pc);
   c=11;
   printf("Address of pointer pc:%d\n",pc);
   printf("Content of pointer pc:%d\n\n",*pc);
   *pc=2;
   printf("Address of c:%d\n",&c);
   printf("Value of c:%d\n\n",c);
   return 0;
}
```

Example 2: pointer to pointer

```
#include <stdio.h>
int main () {
int  var;
int  *ptr;
int  **pptr;
var = 3000;
/* take the address of var */
 ptr = &var;

/* take the address of ptr using address of operator & */

   pptr = &ptr;
   /* take the value using pptr */
   printf("Value of var = %d\n", var );
   printf("Value available at *ptr = %d\n", *ptr );
   printf("Value available at **pptr = %d\n", **pptr);

   return 0;
}
```

Array and pointer

The name of an array holds the memory address of the **first element** of that array.
int Array[5];
int *ptr = &Array;

Pointer	Ptr	Ptr+1	Ptr+2	Ptr+3	Ptr+4
Memory address	1000	1004	1008	1012	1016
Elements	Array[0]	Array[1]	Array[2]	Array[3]	Array[4]
Value/content	2	7	4	5	3

Each integer variable takes 4 bytes. So if an integer is stored at location n, the next one will be stored at n+4.

i) passing array to function using pointer

```
#include<stdio.h>
void printNum(int *ptr, int len);

int main()
{
   int a[4]={4,10,1,5};
   printNum(a,4);
   return 0;
}
void printNum(int *ptr,int len)
{
   int i;
   for(i=0; i <len ; i++ )
   {
      printf("*(ptr+%d) = %d \n",i,*(ptr+i),i);
     // printf("ptr[%d]= %d \n",i, ptr[i]);
   }
}
```

ii) returning pointer from functions

```
#include<stdio.h>
int* getRandom();

int main () {
  int *p;
  p = getRandom();
  printf("address of %d\n",p);
  printf("value at *p : %d\n", *p );

 return 0;
}

int* getRandom( ) {
   static int j;
   j=rand()%10;
   printf("value of j: %d\n",j);
   printf("Address of j: %d\n", &j);

  return &j;
}
```

Key points to remember about pointers in C

1) Normal variable stores the value whereas pointer variable stores the address of the variable.
2) The content of the C pointer always be a whole number i.e. address.
3) Always C pointer is initialized to null, i.e. int *p = null.
The value of null pointer is 0.
4) & symbol is used to get the address of the variable.
5) * symbol is used to get the value of the variable that the pointer is pointing to.
6) If pointer is assigned to NULL, it means it is pointing to nothing.
7) Two pointers can be subtracted to know how many elements are available between these two pointers.

Some Common Mistakes

```
int c, *pc;
pc=c;  //pc is address whereas, c is not an address.
*pc=&c; //&c is address whereas, *pc is not an address.
```

Equivalent Expression

```
list[2]=5;
*(p+2)=5;
 p[2]=5;
 *(list+2)=5;
```

All of the above are same.

Task (10 marks)

1. Write a C program to calculate the area and perimeter of a circle using pointer.
void areaPeri(int *r)
.

2. Write a function which will display sum of two arrays using pointers.

3. Write a C program using pointers to read in an array of integers and print its elements in reverse order.

4. Create an array of size given by the user. Find a number taken as input from user. Your task is to print the memory address and the index of the number using pointer to travel the array. Note that if number is found at multiple position you should print all memory address.