Solution to Exercises in L#26

Solution to Example on Slide 5

- What is the output of the following program?

```
#include "stdio.h"
void main(void)
{
    int x=3;
    int *p= &x;
    printf("%d\n",x);
    printf("%d\n",*p);
    printf("%d\n", p);
}
```

```
3
3
15989752
```

Solution to Exercise on Slide 8

```
#include "stdio.h"
void main(void)
{
    int x=3;
    int *p=&x;
    ++x;
    *p=*p+1;
    (*p)++;
    printf("x is: %d\n",x);
}
```

x is 6

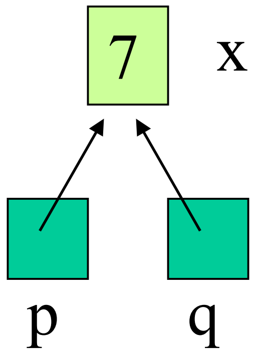
Solution to Exercise on Slide 10

```
#include "stdio.h"
void main(void)
{
    int x = 10;
    int y = 10;
    int *px = &x;
    int *py = &y;
    if(*px == *py)
          printf("The numbers are equal\n");
}
```

Solution to Exercise on Slide 11

```
#include "stdafx.h"
void main(void)
{
    int x = 7;
    int *p= &x;
    int *q = &x;
    printf("*p is %d\n",*p);
    printf("*q is %d\n",*q);
    printf("p is %d\n", p);
    printf("q is %d\n", q);
}
```

```
*p is 7
*q is 7
p is14678392
q is 14678392
```



Solution to Example on Slide 15

```
#include "stdafx.h"

void main(void)
{
    int x = 10;
    int *p;         /*p is a pointer to an integer*/
    p = &x;
    int **q;        /*q is a pointer to an integer pointer*/
    q= &p;
    printf("%d\n", x);
    printf("%d\n", *p);
    printf(“%d\n", **q);
}
```

```
10
10
10
```

To refer to x using q, you have to dereference it twice to get to the integer x because there are two levels of indirection / pointers involved!

Solution to Example on Slide 22

Are there any errors/dangers in this program?

```
#include "stdio.h"
int *max(int *pa, int *pb);
void main(void)
{
      int a;
      int b;

      int *pmax=NULL;
      printf("Enter first number:\n");
      scanf("%d",&a);

      printf("Enter second number:\n");
      scanf("%d",&b);

      pmax=max(&a, &b);
      printf("The maximum is %d\n",
      *pmax);
 }
```

```
int* max(int *pa, int *pb)
{
      int larger;

      if (*pa > *pb)
            larger=*pa;
      else
            larger=*pb;
      printf("The larger one is %d\n", larger);
      return &larger;
}
```

YES! It returns a pointer to a local variable *larger in the called function*; when the function max() terminates, its memory can be used by other parts of the program!

Review Questions I (Slide 24: Solution)

1. Which of the following statement defines and initializes a pointer to the address of an integer variable x?

Answer: d)

a) int *ptr=*x;
b) int &ptr = *x;
c) int *ptr=^x;
d) int *ptr = &x;
e) int &ptr=^x;

2. Assume p is a pointer that points to the variable a, which of the following statements will NOT add 1 to the variable a?

Answer: d), f)

a) a++
b) a+=1;
c) a=a+1;
d) p=p+1;
e) *p=*p+1;
f) *p++;

Note for 2 f): postfix increment ++ has a higher priority than indirection operator *; () are needed to force the dereference to occur before the addition so that we add to the data variable, not to the pointer! That is (*p)++

Review Questions II (Slide 25: Solution)

3. Given the following declarations:

```
int a=5;
int b=7;
int *p=&a;
int *q=&b;
int *r=&a;
```

what is the value of each of the following expressions?

a) ++a; → 6

b) ++(*p); →6

c) --(*q); → 6

d) --b; → 6

e) a++; → 5

f) b--; → 7

g) (*r)++; →5

h) (*q)--; →7