

# Programming for Computerteknologi

## Hand-in Assignment Exercises

Week 5: Structured data and pointers

1. (Text answer) function area calculates and returns the area of a rectangle as an integer. The input rectangle is given as four integer coordinates: x1, x2, y1, y2. Complete the function signature below.

```
1
2
3 _____ ( _____ ) {
4
5     return (x2 - x1) * (y2 - y1);
6 }
```

The completed function signature:

```
int area(int x2, int x1, int y2, int y1) {
```

2. The function increment takes a pointer to an integer and adds 1 to the integer value to which it prints. The function does not return any value. Complete the function signature and function body below, so that the main function prints 6 when executed.

```
1
2
3 _____ ( _____ ) {
4
5
6     _____;
7 }
8
9 int main () {
10     int v = 5;
11     increment(&v);
12     printf("%d", v);
13     return 0;
14 }
```

The function signature and function body is as follows to print 6:

```
void increment (int *p) {
    *p += 1;
}
```

3. (Text answer) Consider the following code.

At the end of the function, what are the values for  $x, y, *xp, *yp$ ? Using pen and paper, draw a diagram (like in the lectures) to explain your answer. Your submission must include your diagram. The following diagram formats are allowed: PDF, JPG and PNG.

```
#include <stdio.h>
```

```
int main(void)
{
    int x;
    int y;

    int *xp;
    int *yp;

    x = 5;
    y = x;

    xp = &x;
    yp = &y;

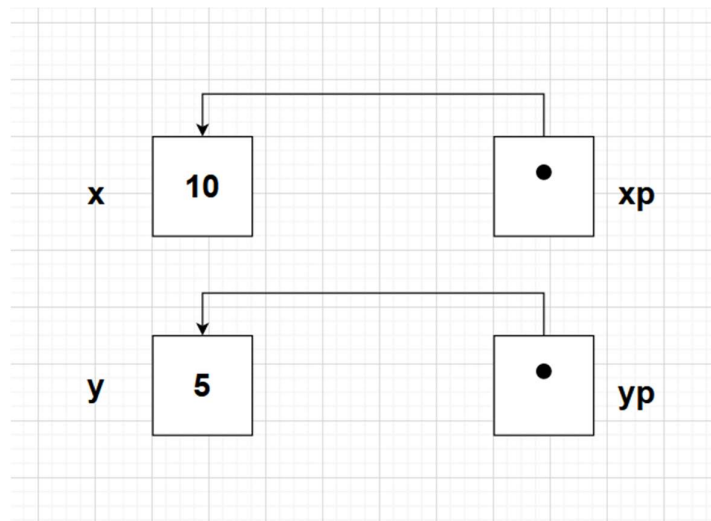
    x = 10;

    /* What are values of: x,y,*xp,*yp */

    printf("x=%d, y=%d, *xp=%d, *yp=%d\n", x,y,*xp,*yp);

    return 0;
}
```

The diagram showing what values the pointers are referring to. The values of  $x, y, *xp, *yp$  are:  
 $x = 10, y = 5, *xp = 10, *yp = 5$



4. (Text answer) Consider the following code.

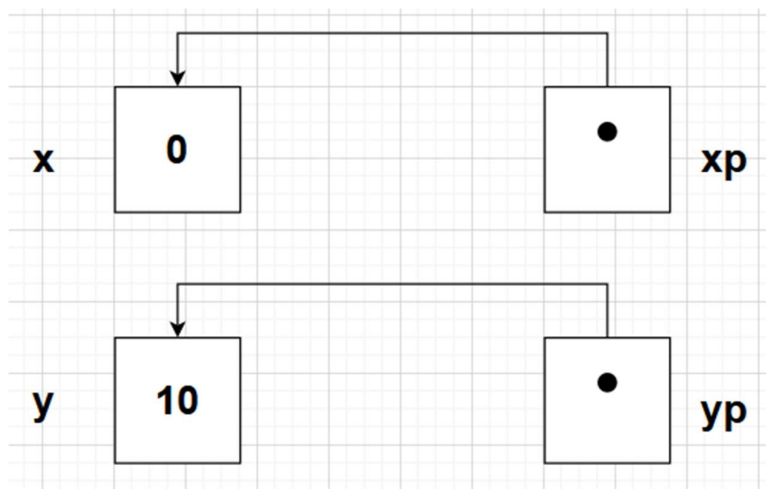
At the end of the function, what are the values for  $x$ ,  $y$ ,  $*xp$ ,  $*yp$ ? Using pen and paper, draw a diagram (like in the lectures) to explain your answer. Remember to include your diagram (in PDF, JPG or PNG format) in your submission.

```
#include <stdio.h>
int main(void) {
    int x;
    int y;
    int *xp;
    int *yp;
    x = 5;
    xp = &x;
    x = 10;
    y = *xp;
    yp = &y;
    *xp = 0;
    /* What are values of: x,y,*xp,*yp */
    printf("x=%d, y=%d, *xp=%d, *yp=%d\n", x, y, *xp, *yp);
    return 0;
}
```

The values of  $x$ ,  $y$ ,  $*xp$ ,  $*yp$  are equal to:

$$x = 0, *xp = 0, y = 10, *yp = 10$$

The diagram shows what values the pointers are referring to:



5. (Text answer) Once again, consider the following code.

At the end of the function, what are the values for  $x, y, *p1, *p2$ ? Using pen and paper, draw a diagram (like in the lectures) to explain your answer. Remember to include your diagram (in PDF, JPG or PNG format) in your submission.

```
#include <stdio.h>

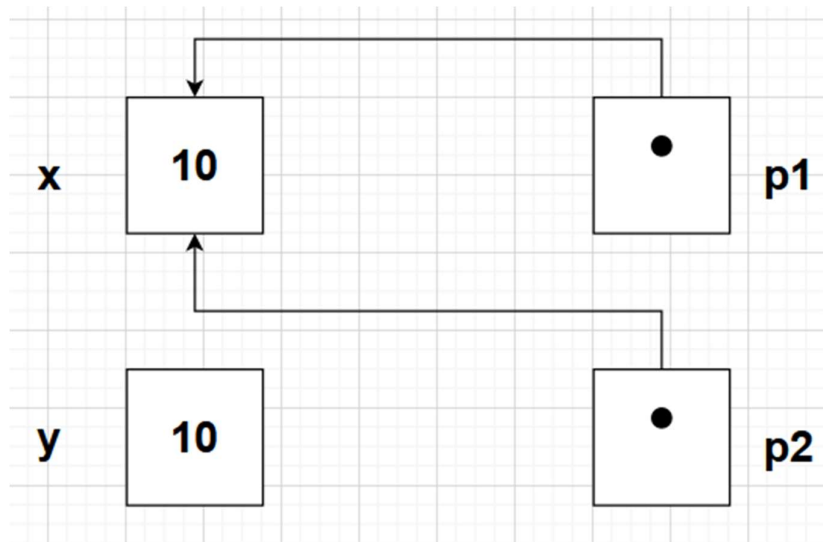
int main(void) {
    int x;
    int y;
    int *p1;
    int *p2;
    x = 5;
    y = 10;
    p1 = &x;
    p2 = p1;
    *p2 = y;
    p1 = &x;

    /* What are values of: x,y,*p1,*p2 */
    printf("x=%d, y=%d, *p1=%d, *p2=%d\n", x,y,*p1,*p2);
    return 0;
}
```

The values of  $x, y, *xp, *yp$  are equal to:

$x = 10, *xp = 10, y = 10, *yp = 10$

The diagram shows what values the pointers are referring to:



6. (Code answer). In the lecture we discussed.

How to represent a geometric point using a C struct. Let's now consider a geometric circle: a circle consists of three integers: x coordinate of the centre point, y coordinate of the centre point, and a radius.

7. (Code answer).

- (a) A sequence of  $n > 0$  integers is called a *jolly jumper* if the absolute values of the differences between successive elements take on all possible values 1 through  $n - 1$ . For instance, 1 4 2 3 is a jolly jumper, because the absolute differences are 3, 2, and 1, respectively. As another example, 11 7 4 2 1 6 is a jolly jumper, because the absolute differences are 4, 3, 2, 1, 5 (the order of the differences does not matter). The definition implies that any sequence of a single integer is a jolly jumper. Write a function to determine whether a sequence is a jolly jumper. The function should have the following signature: `int isJollyJumper(const int seq[], int size)` (Hint: use a boolean array, e.g. `bool diffs_found[n]` to keep track of the differences found so far between consecutive numbers. So that `diffs_found[2]` being `true` implies that the absolute difference 2 has already been found.
- (b) Write a test program that reads the size and sequence, and uses the function to print out if the sequence is a JollyJumper or not.:  
**Input** A line of input contains an integer  $n < 100$  followed by  $n$  integers representing the sequence.  
**Output** For the line of input generate a line of output saying "Jolly" or "Not jolly".