GitHub Link: https://github.com/Aarhus-University-ECE/assignment-5-Daniel6702.git

(1)

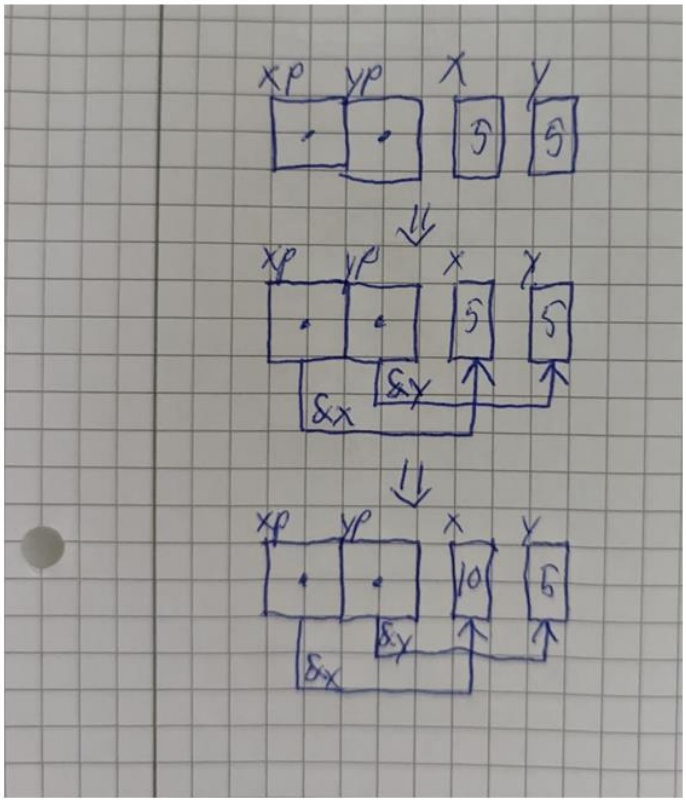
```c
int area(int x1, int y1, int x2, int y2) { //return_type (int), name (area), parameters (integers
    return (x2-x1) * (y2-y1);
}
```

(1)

```c
void increment(int *pointer) {
    (*pointer)++; //(get value at pointer), then increment
}

int main(void) {
    int v = 5;
    increment(&v);
    printf("%d\n",v);
    return 0;
}
```

(2)

```c
int x;
int y;

int *xp;
int *yp;

x = 5;
y = x;

xp = &x;
yp = &y;

x = 10;
```

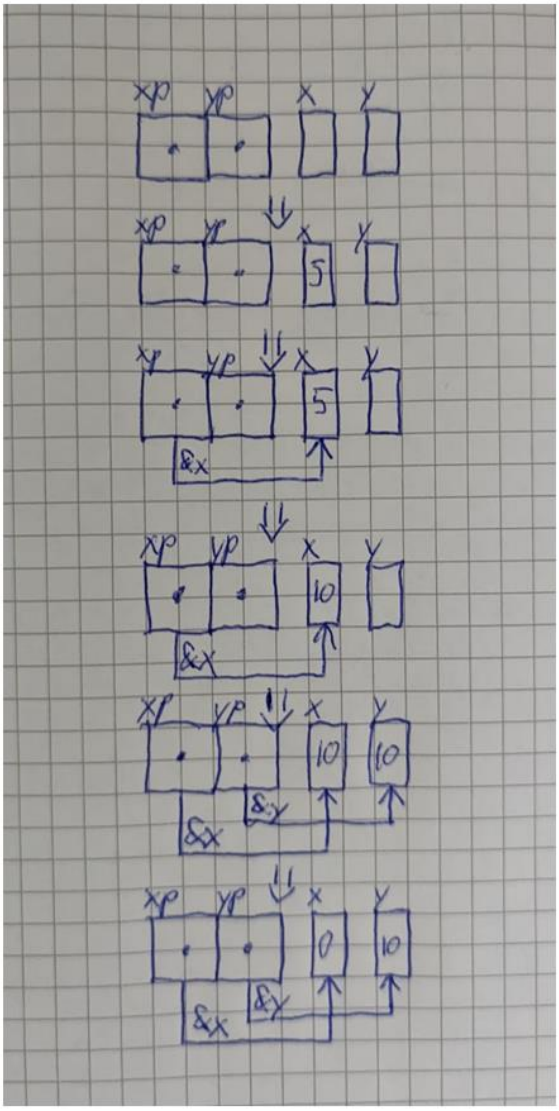


x	y	*xp	*yp

10	5	10	5

(3)

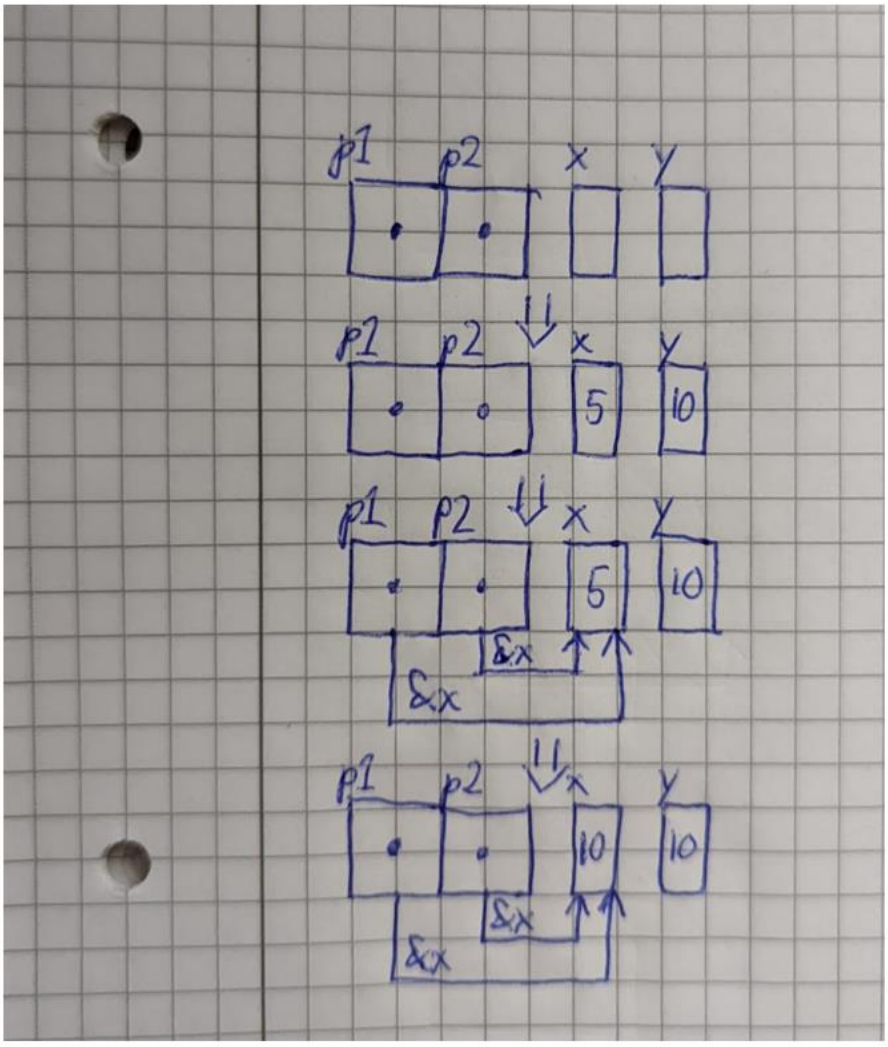
```
int x;
int y;

int *xp;
int *yp;

x = 5;

xp = &x;

x = 10;

y = *xp;

yp = &y;

*xp = 0;
```



x	y	*xp	*yp
0	10	0	10

(4)

```
int x;
int y;

int *p1;
int *p2;

x = 5;
y = 10;

p1 = &x;
p2 = p1;

*p2 = y;

p1 = &x;
```



x	y	*p1	*p2
10	10	10	10

(5)

(a)

```c
typedef struct { //point struct
    int x;
    int y;
} point;

typedef struct { //Circle struct
    point p;
    int r;
} circle;
```

(b)

```c
void fiveCircles(circle c[]) {
    for (int i = 0; i < 5; i++) {
        c[i].r = i;
        c[i].p.x = i;
        c[i].p.y = i;
    }
}
```

(c)

```c
int circleIsValid(const circle *c) {
    if ( (*c).r > 0 ) { //get value (circle struct) at pointer, if radius of circle > 0
        return 1; //return true (1) if circle is valid
    } else {
        return 0;
    }
}
```

(d)

```c
void translate(circle *c, const point *p) {
    (*c).p.x += (*p).x; //adds coords of the two points
    (*c).p.y += (*p).y;
}
```

(6)

(a)

Method 1

```c
int isJollyJumber(const int seq[], int size) {
    bool diffs_found[size-1]; //create array, elemnt is true if difference for coresponding index is found
    for (int i = 1; i < size; i++) {
        diffs_found[abs(seq[i-1]-seq[i])-1] = true; //calculating absolute differences between succesive elements
    }                                               //and assigns true for that index.
    for (int i = 0; i < size-2; i++) {
        if (diffs_found[i]!=true) {   //if an element in diffs_found isnt true, cant be jolly
            return false;
        }
    }
    return true; //if all true, jolly
}
```

Method 2

```c
int isJollyJumber(const int seq[], int size) {
    /*We can predict what the sum of the differences should be, 1+2+3...+n
      with the following formula: n(n+1)/2
    */
    int sum = 0;
    int predicted_sum = (size-1)*size/2;
    for (int i = 1; i < size; i++) {
        sum += abs(seq[i-1]-seq[i]); //summing up the absolute differences between succesive elements
    }
    return sum == predicted_sum ? 1 : 0;  //using ternary operator to return 1 or 0
}
```

(b)

Test program, user inputs length and sequence

```c
int main(void) {
    int n = 0;
    int seq[n];
    printf("Enter sequence length, then the sequence:\n");
    scanf("%d",&n); //Input length
    for (int i = 0; i < n; i++) {
        scanf("%d",&seq[i]); //Input sequence
    }
    int result = isJollyJumber(seq,n); //calls function
    (result == 1) ? printf("jolly") : printf("not jolly"); //prints the result
    return 0;
}
```