

ENSF 337 Tutorial 2 – Tuesday Sept 26 2018

M. Moussavi

Problem I: Draw AR diagrams when function `double_trouble` reaches point one for the **first time** and then for **the second time** (2 separate diagrams).

```
#include <stdio.h>
void double_trouble(int *p, int y);
void trouble(int *x, int *y);

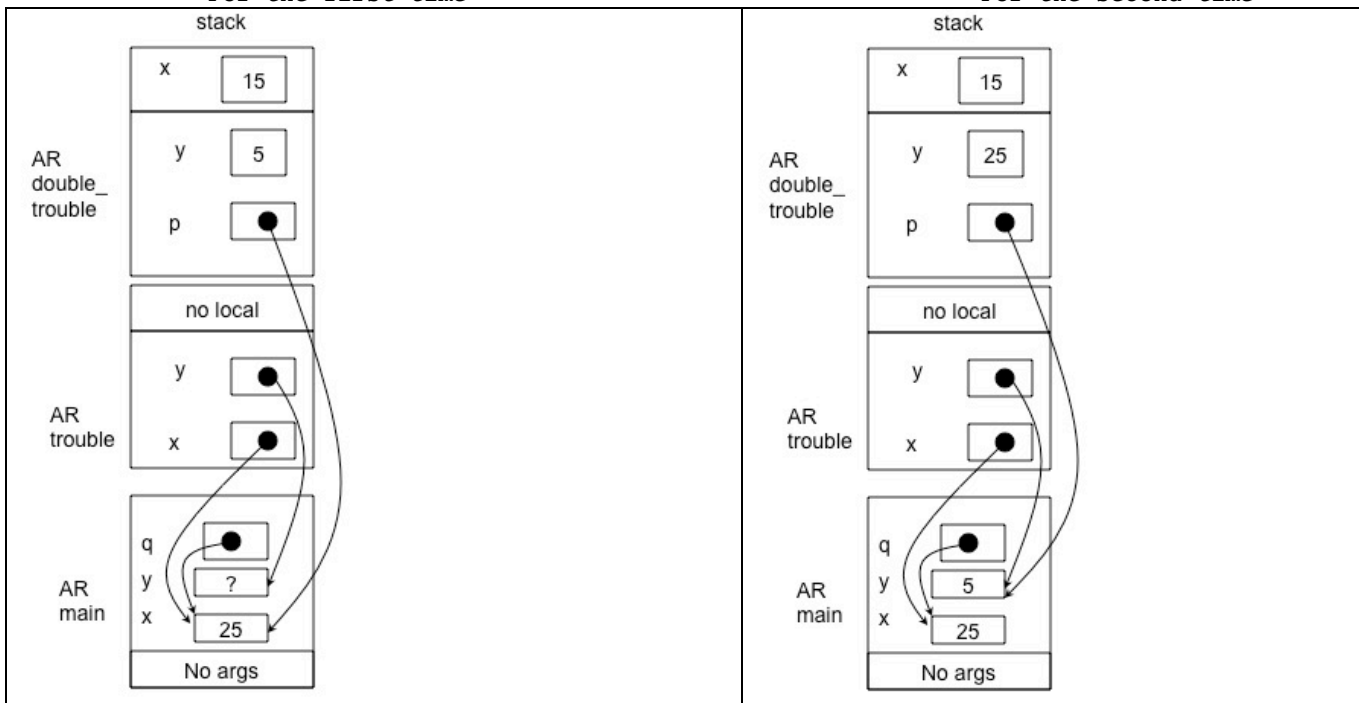
int main(void){
    int x , y ;
    int *q = &x ;
    trouble(q, &y);
    return 0;
}

void double_trouble(int *p, int y){
    int x;
    x = 15;
    *p = 2 *x -y;
    // point one
}

void trouble (int *x, int *y){
    double_trouble(x, 5);
    double_trouble(y, *x);
}
```

For the first time

For the second time



Problem II: Write a function that matches the following function interface comment:

```
char* clean(char *s, int c);
/* REQUIRES: s points to the beginning of a string. c contains a character code.
 * PROMISES: to remove all occurrences of c in s and returns the result
 * EXAMPLE USES:
 * char s[] = "banana";
 * clean(s, 'b') should return anana.
 */
```

One Possible solution:

```
char* clean(char* str, char c) {
    int i=0, j =0, n = 0;
    while (str[n++]);
    while(i<n )
    {
        if(i != j) str[j] = str[i];
        if(str[i] != c) j++;
        i++;
    }
    str[j] = '\0';
    return str;
}
```

Another Possible solution:

```
char* clean(char* str, char c) {
    int i = 0, j =0;
    while(str[i])
    {
        if(i != j) str[j] = str[i];
        if(str[i] != c) j++;
        i++;
    }
    str[j] = '\0';
    return str;
}
```

Another Possible solution:

```
char* clean(char* str, char c) {
    int i = 0, j =0;
    int n = (int)strlen(str);
    while(i<n )
    {
        if(i != j) str[j] = str[i];
        if(str[i] != c) j++;
        i++;
    }
    str[j] = '\0';
    return str;
}
```

Problem III: Write the definition of function `str_to_num` as indicated in the following function interface comment. You are not allowed to use any C Library function.

```
int str_to_num (const char *s);
/* REQUIRES: s is pointing to a built-in string that terminates with '\0'.
 * Characters in s must be all digits
 * PROMISES: returns an integer value equivalent to the string of digit.
 * EXAMPLE:
 *   if s [0]== '8', s[1] == 5, s[2] == 3, and s[4] == '\0', the return value must be 853.
 */
```

```
int str_to_num ( const string s)
{
    int num =0;
    int i =0;
    while(i < s.length()){
        if(s[i] < '0' || s[i] > '9')
            return -1;

        num = num * 10 + (s[i] - 48);
        i++;
    }
    return num;
}
```

```
int main()
{
    const char* s = "4563";
    int y = str_to_num(s);
    printf("%d\n", y);    // prints  4563 as an integer

    return 0;
}
```

Problem IV: Write the definition of the following function. You are not allowed to use any C Library function.

```
int first_occurance(const char* str, const char ch);
/* REQUIRES: str pointing to a built-in string terminated with '\0'
 * PROMISES: returns -1 if ch doesn't exist in str. Otherwise, returns the position
 * of the first occurrence of ch in str.
 */
```

```
int first_occurance(const char* str, const char ch)
{
    int i;
    for(i = 0; str[i] != '\0'; i++)
        if(str[i] == ch) return i;

    return -1;
}
```