Metropolia University of Applied Sciences

Programming

TI00AA43-3003

Lecture 5

Sami Sainio

sami.sainio@metropolia.fi

Plan for going forward

- Variables and printing to screen
- If, else, while and for loops
- I/O
- Functions and tables
- File handling
- **Pointers and arrays**
- Simple structures
- Program structure and design

6. Lectures

Pointers and arrays

Array revision:

- Array, meaning a collection of variables, that are indexed by integers
- Example: int tbl[5]; defines an array ”tbl” that has five (5) elements that are of int type
- Elements are referred as: tbl[0], tbl[1], tbl[2], tbl[3] and tbl[4]
- NOTE! Indexing starts at 0 and ends at size-1
- Elements of the array act as variables of the defined type

```
#include <stdio.h>
#define MAXNUMBERS 10
int main(void) {
        int nr, i;
        int counter = 0;
        int sum= 0;
        int allNum[ MAXNUMBERS ];
        do {
                printf(”Give integer: (negative ends): ");
                scanf("%d", &nr);
                if(nr >= 0) {
                allNum[ counter++ ] = nr;
        }
        } while ((nr>= 0) && (counter < MAXNUMBERS));
        for(i = 0; i < counter; i++) {
                sum = sum + allNum[ i ];
        }
        printf(“Thanks. Sum is %d\n", sum);
        return 0;
}
```

Pointers – address and indirect operators

- Operator & in front of the variable returns the variables ***address***
- Operator * in front of the address returns the variabes contents (value saved to the variable)

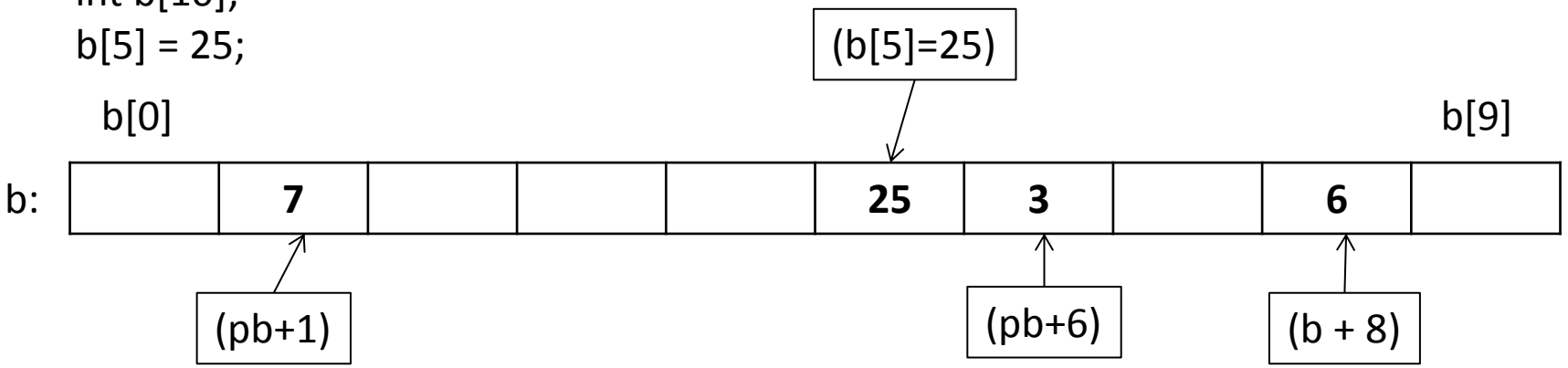
```
int a = 1, int b = 2;
int *addr;                  // addr  is pointer to int type variable

addr = &a;                  // pointer is given variable a's address
b = *addr;                  // b == 1
*addr = 0;                  // a == 0
```

Observations from pointers

- Define pointers: int *pointa, *pointb, a;
  - For example pointer *pointa can be used anywhere, where int type is used:
  - *pointa = *pointa + 5; a = *pointa +1; *pointa += 1; ++(*pointa); (*pointa)++;
  - Pointers can be used for giving value, compare etc.
  - pointb = pointa; if (pointa == pointb) …

Pointers and tables

```
int b[10];
b[5] = 25;
```




```
int * pb;
pb = &b[0];          // pb points to b[0]
pb = b;              // as previous, pb points to b[0]
*(pb + 1) = 7;       // b[1] == 7
*(pb + 6) = 3;
b[ 8 ] = 6; /* OR */ *(b + 8) = 6;          // C-compiler changes: b[i] -> *(b + i)
```

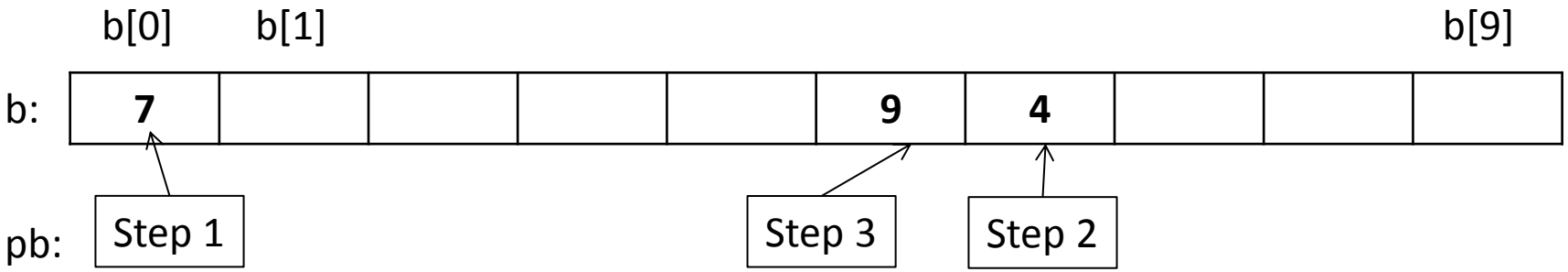
NOTE!

Remember that tables name is NOT a variable, meaning that b++ will not work!

b = pb; will not work neither!

Pointer arithmetics

```
int b[ 10 ];
int *pb;
pb = b;              // Step 1
*pb = 7;             // set b[ 0 ] == 7
pb = pb + 6;         // Step 2
*pb = 4;
pb--;                // Step 3
*pb = 9;
```



Reading a string – fgets()

```
#include <stdio.h>
#include <string.h>                // Note! New library!

#define MAXLINELEN 10
int main(void) {
        char line[ MAXLINELEN ];
        int i;
        printf(”Input a string: ");
        fgets(line, MAXLINELEN, stdin);          // fgets(char *addr, int siz,  FILE *stream)
        printf("String is: %s\n", line);
        for(i = 0; i < strlen(line); i++) {      //strlen returns string lenght,
                                                 //including \n
                printf(”Character %c, ASCII %d\n", line[ i ], line[ i ]);
        }

return 0;
}
```