

IS12 - Introduction to Programming

Lecture 14: Character Processing

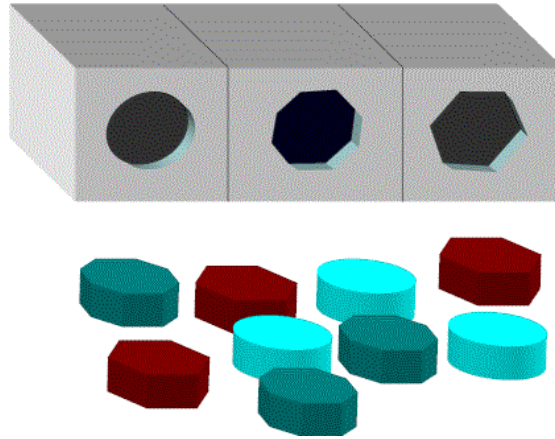
Peter Brusilovsky

<http://www2.sis.pitt.edu/~peterb/0012-051/>

Outline

- Characters and operations
- Input/output redirection
- Simple file processing
- Advanced File processing

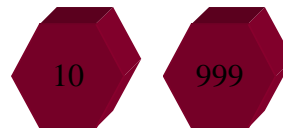
Primitive Data Types



Primitive Data Types

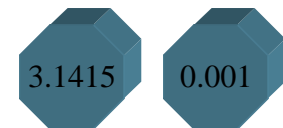
- Integer data

- ◆ 1, 10, 999, 1000



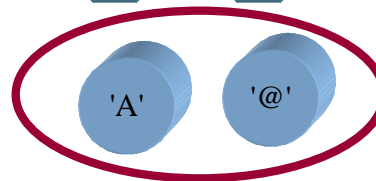
- Floating point data

- ◆ 3.1415, 0.001, 2.0



- Characters

- ◆ 'A', 'B', '_', '@'





Characters – Another Data Type

- Smallest storage space – one byte
- Stores code of one character (0-255)
 - printable characters as 'a' or '@' or ''
 - non-printable characters '\0', '\n'
- C programs can read characters

```
scanf("%c", &mychar);
```

or simpler: `mychar = getchar();`
- C programs can print characters

```
printf("%c", mychar);
```

or simpler: `putchar(mychar);`



Typecasting - Explicit Conversion

- Types of data type conversion:
 - Implicit: arithmetic operations, assignment
 - Explicit: typecasting
- Typecasting operator:

```
(datatype) expression
```
- Examples:

```
(float) 3 / 2 ⇨ 3.0 / 2 ⇨ 1.5  
(int) (10.0 / 3) ⇨ 3  
(int) 'a' ⇨ ???
```



Example: Character Codes

```
#define FIRSTCHAR ' '  
#define LASTCHAR 'z'  
#include <stdio.h>  
  
void main () {  
    char c = FIRSTCHAR;  
    while(c <= LASTCHAR) {  
        printf("Code for %c is %d\n", c, (int) c);  
        ++c;  
    }  
}
```



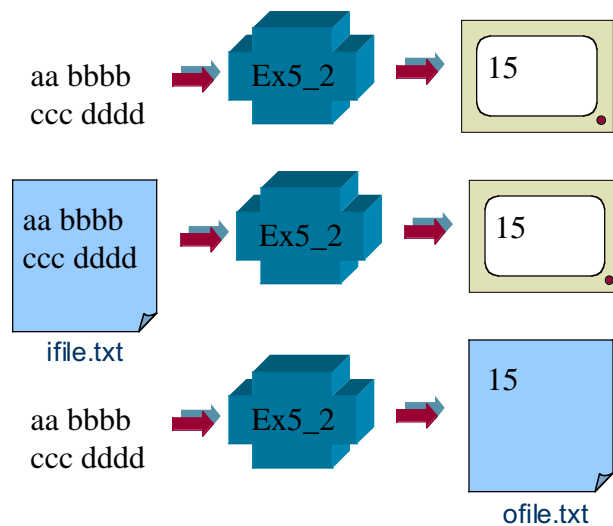
Example: Counting Characters

```
#include <stdio.h>  
  
void main () {  
    long nc;  
  
    nc = 0;  
    while(getchar() != EOF)  
        ++nc;  
    printf("%ld\n", nc);  
}
```

Standard Input and Output

- Each program has two streams - standard input and standard output
- By default input is connected to the keyboard and output to the screen
C:\> Ex5_3
- But it could be redirected
 - Input from file: C:\> Ex5_3 <infile.txt
 - Output to file: C:\> Ex5_3 >ofile.txt

Stream Redirection





Example: File Copying

```
#include <stdio.h>

void main () {
    int c;

    c = getchar();
    while(c != EOF) {
        putchar(c);
        c = getchar();
    }
}
```

- What is about EOF?
- EOF is not a real symbol, it is not even char type, it is int (that's why int c;)
- getchar() is a request to the *operating system* for the next symbol
- if there are no more symbols, the operating system has to tell about it to the program - it returns this special value EOF



Example: Line Counting

```
#include <stdio.h>

void main () {
    int c, nl;

    nl = 0;
    while((c = getchar()) != EOF)
        if(c == '\n')
            ++nl;
    printf("%d\n", nl);
}
```



Before next lecture:

- Do reading assignment
- Perry: Chapter 10: Typecasting;
Chapter 18
- Use KnowledgeSea to find more readings
- Run Classroom Examples
 - This time it is VERY important