

Computer Programming: Skills and Concepts

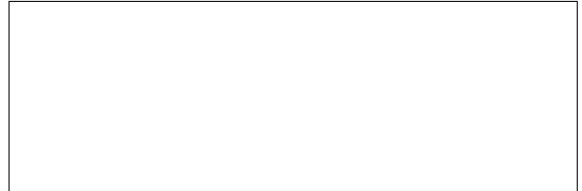
Tutorial 5 (Tue 1 Nov – Fri 4 Nov)

Arrays

What is the value of the array `a` at the indicated point?

```
void Rev(int a[], int b[], int n) {
    int i;
    for (i=0; i < n; i++) { b[n-1-i] = a[i]; }
}
```

```
int main(void) {
    int a[4] = { 0, 1, 2, 3 };
    int b[4];
    Rev(a,b,4);
    Rev(b,a,3);
    /* what is the value of a here? */
}
```




Another function

We would like to have a function that takes ($n = 3$) three numbers a_i and gives us the average μ and the variance σ^2

$$\mu = \frac{1}{n} \sum_{i=1}^n a_i$$
$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (a_i - \mu)^2$$

How can we write a function that returns two values?



I/O with characters

Consider the following code:

```
#include <stdio.h>
int main(void) {
    int c;
    while ((c = getchar()) != EOF) {
        printf("char %c, ASCII code %d\n", c, c);
    }
}
```

What gets printed on the screen for the following input: 0123 abc ABC

Programming

In lectures 10–11 we wrote a program to print numbers in different bases. Here is (a slightly shortened version of) the finished program (available as `base.c` from the course page):

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

void PrintDigit(int d) {
    printf("<%d>",d);
}

int main(void) {
    int n,b;
    printf("Please give me n and b: ");
    scanf("%d %d",&n,&b);

    if ( n < 0 ) { printf("-"); n = n*-1; }
    int bp;
    bp = 1;
    while ( bp*b <= n ) { bp = b*bp; }
    int digit;
    while ( bp > 0 ) {
        digit = n/bp;
        PrintDigit(digit);
        n = n % bp;
        bp = bp/b;
    }
    printf("\n");
    return EXIT_SUCCESS;
}
```

How can we extend the program so that the number `n` is a floating point number. Assume that the representation should be printed to exactly `PRECISION` places after the ‘decimal’ point. For example, with `PRECISION == 6`, decimal 6.25 should be printed in base 4 as 12.100000, and decimal 1.5 should be printed in base 3 as 1.111111. **Note:** you can get the integer part of a *non-negative* floating point number by casting it to `int` (as long as the number isn’t too big to be an `int`).

(Optional!) Work out how to avoid printing a string of trailing zeros, so that 6.25 in base 4 is printed as 12.1.

After discussing this in the tutorial, implement the solution in your own time.