

A Worked Example

Consider the problem of making change with US coins for a specific amount. The goal is a C program to do this. But first we introduce constants.

3.1 Constants

If you have the same fixed value recurring throughout the program, you should use a constant at the top of the program. The “new” C style is to use the reserved word `const`:

```
const int NUM_STATES = 50;
```

For the rest of the program, `NUM_STATES` is just a normal variable, except that its value cannot be changed. It is common style to use uppercase for constants.

3.2 A Program for Making Change

With any programming task, you should think how **you** would do it (as a human computer). And then plan the steps.

In this case, the key question is what recipe (what computer science calls an **algorithm**) should be used for making change. But cashiers know: take as many quarters as are at most the amount; then as many dimes so that still at most the amount; and so on.

The first plan for a program might be:

- 1) Input number
- 2) Calculate number of quarters
- 3) Calculate number of dimes
- 4) Calculate number of nickels
- 5) Calculate number of pennies
- 6) Output results

Calculating the number of quarters uses integer division. How?

Suppose input is stored in `amount`, an `int` variable. Then the number of quarters is

```
amount / 25
```

And you need to make change for the remainder, which is

```
amount % 25
```

And now for a program, called `makeChange.c`

```
/* making change with coins
 * wdg 2008
 */
#include <stdio.h>

const int QUARTER = 25;
const int DIME = 10;
const int NICKEL = 5;

int main( void ) {
    int amount, quarters, dimes, nickels, pennies;

    printf("Enter amount of change: ");
    scanf("%d", &amount);

    quarters = amount / QUARTER;
    amount = amount % QUARTER;
    dimes = amount / DIME;
    amount = amount % DIME;
    nickels = amount / NICKEL;
    pennies = amount % NICKEL;

    printf("%d quarters, %d dimes, %d nickels, %d pennies\n",
           quarters, dimes, nickels, pennies);

    return 0;
}
```

Note that the call to `printf` is spread over two lines; most white space is just there for the human reader and is ignored by the compiler. Strings are an exception: they cannot be broken over a line.