#include <stdio.h>
define MAXLINE 1000 /* maximum input line size */

int getline(char line[], int maxline);
void copy(char to[], char from[]);

/* print longest input line */
main()
{
    int len; /* current line length */
    int max; /* maximum length seen so far */
    char line[MAXLINE]; /* current input line */
    char longest[MAXLINE]; /* longest line saved here */

    max = 0;
    while ((len = getline(line, MAXLINE)) > 0)
    {
        if (len > max) {
            max = len;
            copy(longest, line);
        }
        if (max > 0) /* there was a line */
            printf("%s", longest);
    return 0;
}

/* getline: read a line into s, return length */
int getline(char s[], int lim)
{
    int c, i;

    for (i=0; i<lim-1 && (c=getchar())!=EOF && c!='\n'; ++i)
        s[i] = c;
    if (c == '\n') {
        s[i] = c;
        ++i;
    }
    s[i] = '\0';
    return i;
}

/* copy: copy 'from' into 'to'; assume to is big enough */
void copy(char to[], char from[])
{
    int i;

    i = 0;
    while (((to[i] = from[i]) != '\0')
        ++i;
}
FORMATTED INPUT — scanf (Contd)

scanf (format [, pointer — list])

FORMAT:

QUOTED (") STRING CONTAINING:

- WHITE SPACE
- PLAIN CHARACTERS (OPTIONAL)
- CONVERSION SPECIFICATIONS

POINTER,LIST:

DESTINATIONS FOR INPUT ITEMS
**scanf()**

**NAME**

scanf

**SYNOPSIS**

```c
#include <stdio.h>
int scanf(format[,]pointer list])
```

**DESCRIPTION**

Reads characters from *stdin* according to *format*. Stops on first conflict, offending character left unread. Stores results at addresses in pointer list. Returns number of %’s matched, EOF on end-of-file.

%conversion-char

- **c** any 1 character
- **d,u,o,x,X** integer: decimal, unsigned, octal, hex precede with l if long, h if short
- **e,f** float, preceded with l if double
- **s** string of non-whites, \'\0\' added to destination string or array

**EXAMPLE**

```c
1 #include <stdio.h>
2 main()
3 {
4   int ret, num;
5
6   printf("Please enter an integer: ");
7   ret = scanf("%d", &num);
8   ...
```
```c
scanf() - error recovery

1 /* Unsuccessful attempt to force */
2 /* user to enter valid input */
3 #include <stdio.h>
4 main()
5 {
6    int    age;
7
8    printf("Enter age: ");
9    while (scanf("%d", &age) != 1)
10    printf("Try again. Age: ");
11    printf("Thank you. Age is %d. \n", age);
12    ...

Terminal Screen:

Enter age: Why?
Try again. Age: Try again. Age: ...
```
scanf() - error recovery, continued

- Clear to end of line, field, or record
- Or exit the program

```c
/* Forces user to enter valid input */
#include <stdio.h>
main()
{
  int age;
  printf("Enter age: ");
  while (scanf("%d", &age) != 1) {
    while (getchar() != '\n') ;  /* Clear line */
    printf("Try again. Age: ");
  }
  printf("Thank you. Age is %d.\n", age);
```

**Terminal Screen:**

Enter age: Why?
Try again. Age: 42
Thank you. Age is 42.
The digits at the left of the table are the left digits of the decimal equivalent (0-127) of the character code, and the digits at the top of the table are the right digits of the character code. For example, the character code for 'F' is 70, and the character code for '&' is 38.