Programs must be created in files whose names end in `.c'. The form of a program is:

```c
#include <stdio.h>

/* Stuff enclosed in these things is a comment */

int main (int argc, char **argv)
{
    /* Variable declarations here */
    /* Executable statements here */
    return 0;
}
```

Create the following program in file `first.c'.

```c
int main (int argc, char **argv)
{
    int  ii;
    ii = 5;
    printf ("The value of ii is %d\n", ii);
    if (argc > 1)
        printf ("Argument 1 is \"%s\n", argv[1]);
    else
        printf ("No arguments were passed except the command name \"%s\n", argv[0]);
    return 0;
}
```

Compile the program with
```
cc first.c -o first
```

Run the program with
```
first
```

The most commonly used types of storage cells are:

- `char /* stores a character */`
- `int /* stores an integer */`
- `float /* stores a real */`

**Constants**

- `22`
- `022`
- `‘c’`
- `‘\t’ /* TAB */`
- `‘\0’ /* NEWLINE */`
- `5.2`
- `.2`
- `5.
- `5e4`
- `5e+4`
- `5e-4`
- `5.2e9`
- `.2e9`
- `5.e9`
Statements

```c
var = exp;

if (exp)
{
    stmt list
}
else
{
    stmt list
}

while (exp)
{
    stmt list
}
break;
continue;
exit(exp);
```

operators

```c
- a /* unary minus */
! a /* logical not */
a * b /* mult */
a / b /* div */
a % b /* remainder or mod */
a + b /* add */
a - b /* sub */
a < b /* less than */
a > b /* greater than */
a <= b /* less than equal */
a >= b /* greater than equal */
a != b /* inequality */
a == b /* equality */ (IMPORTANT NOTE: the equality operator is ‘==’, not ‘=’ !!!!!!!)
a && b /* logical and */
a || b /* logical or */
```

I/O

```c
#include <stdio.h>
/* stdin, stdout, stderr */

putchar(chr);
chr = getchar();

fprintf(stream, format, a1, a2, ...);

printf(format, a1, a2, ...);
/* same as fprintf (stdout, format, ...) */

fscanf(stream, format, &a1, &a2, ...);
scanf(format, &a1, &a2, ...);
```
The format string is a string with "variable specifiers" marking where variable values are applied. The form of a variable specified is

\%W.PT

where

- L \text{ef} \text{justify (seldom used).}
W \text{Minimum field width (a larger width will be used if necessary).}
P \text{Precision (seldom used). The number of digits after the decimal point for real types, the number of digits for integer types (leading zeros added as required).}
T \text{One of the following types:}
\begin{itemize}
\item d \text{integer decimal}
\item o \text{integer octal (unsigned)}
\item x \text{integer hexadecimal (unsigned)}
\item f \text{float or double real}
\item e \text{float or double exponential}
\item c \text{character}
\end{itemize}

examples:

\begin{verbatim}
printf ("This is \%c format string \%-5d\n", 'a', ii);
printf ("Look at the leading zero %.6d\n", 12);
printf ("Look at the precision %.6f\n", 12.2);
\end{verbatim}