Format specifiers for simple variable types

<table>
<thead>
<tr>
<th>Data types</th>
<th>printf conversion specifications</th>
<th>scanf conversion specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>long double</td>
<td>%Lf</td>
<td>%Lf</td>
</tr>
<tr>
<td>double</td>
<td>%f</td>
<td>%f</td>
</tr>
<tr>
<td>float</td>
<td>%f</td>
<td>%f</td>
</tr>
<tr>
<td>unsigned long int</td>
<td>%lu</td>
<td>%lu</td>
</tr>
<tr>
<td>long int</td>
<td>%ld</td>
<td>%ld</td>
</tr>
<tr>
<td>unsigned int</td>
<td>%u</td>
<td>%u</td>
</tr>
<tr>
<td>int</td>
<td>%d</td>
<td>%d</td>
</tr>
<tr>
<td>unsigned short</td>
<td>%hu</td>
<td>%hu</td>
</tr>
<tr>
<td>short</td>
<td>%hd</td>
<td>%hd</td>
</tr>
<tr>
<td>char</td>
<td>%c</td>
<td>%c</td>
</tr>
</tbody>
</table>

simple variable types

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size*</th>
<th>Range*</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>Character or small integer.</td>
<td>1byte</td>
<td>signed: -128 to 127</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>unsigned: 0 to 255</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>Short Integer.</td>
<td>2byte</td>
<td>signed: -32768 to 32767</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>unsigned: 0 to 65535</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>long int</td>
<td>Long integer.</td>
<td>4byte</td>
<td>signed: -2147483648 to 2147483647</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>unsigned: 0 to 4294967295</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td>Floating point number.</td>
<td>4byte</td>
<td>+/- 3.4e +/- 38 (~7 digits)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>double</td>
<td>Double precision floating point number.</td>
<td>8byte</td>
<td>+/- 1.7e +/- 308 (~15 digits)</td>
</tr>
</tbody>
</table>
Write a program to calculate the volume of a cylindrical tank.

Let's remember the algorithm:

1. Get r as radius and h as height.
2. Calculate \( V = r \cdot r \cdot h \cdot 3.14 \).
3. Show the result.

```c
#include "stdio.h"

int main() {
    float r, h, pi, V;
    pi = 3.14;
    printf("radius? ");
    scanf("%f", &r);
    printf("height? ");
    scanf("%f", &h);
    V = pi * r * r * h;
    printf("\n\nVolume is \%f", V);
    return 0;
}
```
In order to see last screen at the end of program running, we add line 2 and 12. Details will given later

```c
#include "stdio.h"
#include "conio.h"

int main () {
    float r, h, pi, V;
    pi=3.14;
    printf("radius ? ");
    scanf("%f", &r);
    printf("height? ");
    scanf("%f", &h);
    V=pi*r*r*h;
    printf("\n\nVolume is %f", V);
    getch();
    return 0;
}
```

A variable can get first value in declaration part

```c
#include "stdio.h"
#include "conio.h"

int main () {
    float r, h, V, pi=3.14;
    printf("radius ? ");
    scanf("%f", &r);
    printf("height? ");
    scanf("%f", &h);
    V=pi*r*r*h;
    printf("\n\nVolume is %f", V);
    getch();
    return 0;
}
```
A variable can be declared as a constant first value in declaration part

```c
#include "stdio.h"
#include "conio.h"

int main () {
    float r,h,V;
    const float pi=3.14;
    printf("radius ? ");
    scanf("%f",&r);
    printf("height? ");
    scanf("%f",&h);
    V=pi*r*r*h;
    printf("\n\nVolume is %f",V);
    getch();
    return 0;
}
```

One scanf can take several values at a time

```c
#include "stdio.h"
#include "conio.h"

int main () {
    float r,h,V;
    const float pi=3.14;
    printf("radius,height ? ");
    scanf("%f,%f",&r,&h);
    V=pi*r*r*h;
    printf("\n\nVolume is %f",V);
    getch();
    return 0;
}
```
Write a program to convert seconds to hours:minute:seconds equivalent

Example

```c
#include<stdio.h>
#include"conio.h"
int main(){
clrscr();

int seconds;
int h,m,s,r;// r->residue
printf("enter the seconds ");
scanf("%d",&seconds);
h=seconds/3600;
r=seconds-h*3600;
m=r/60;
s=r-m*60;
printf("%d seconds = %d hours : %d minutes : %d seconds",seconds,h,m,s);
getch();
return 0;
}
```
Example

A bank customer wants to withdraw money from ATM. Available types of notes are 100, 50, 20, 10, 5.

Write a program of money withdrawing to determine how much of that is paid with what type of notes.

For example 636 TL

6x100+0x50+3x10+1x5 and 1 can not be paid