

Introduction

Vs code'da terminalda command yazmaq – mousedan istifadeni azaltmaq. CLI – command line interfacedir.

```
1. $ code hello.c
```

– commandi hello adli C file'i yaradir.

```
1. $ make hello
```

Commandi hello.c adli file'i machine code'a ceviris.

```
1. ./hello
```

Execute edir kodu. Eger ./hello.c yazsan execute etmeyecek, cunki source code executable deyil.

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

<stdio.h> header file'idir hansiki librarydir, ve bezi commandlari bu headeri elave etdikden sonra istifade etmek olur.

```
#include <stdio.h>
#include <cs50.h>

int main(void)
{
    string answer = get_string("What's your name? "); #buradaki string cs50nin
    funksiyasidir.
    printf("hello, %s\n", answer);
}
```

Yuxaridaki kodda, {variable} elemek ucun C-de % istifade edirik s herfi stringin bas herfini ifade edir, sonraki \n ise qoyulur ki, terminalda novbeti commandi yazanda command prompt yeni dollar isaresi (\$), yaziya yapismasin

```
$ make hello
$ ./hello
What's your name? nargiz
hello, nargiz
$ make hello
$ ./hello
What's your name? nargiz
hello, nargiz$
```

- C'de char type isletsen single quote, string isletsen ise double quote isletmelisen.

For killing terminal **ctrl+c**. bezen infinite loopa girende lazim ola biler.

Functions

Funksiyani kodun altına yazanda, execute ede bilmir, cunki kod funksiyanın var olmasından evvel icra olunur. Bunun ucuñ funksiyanın ilk setrini kopyalayıb file'in evveline yapisdirmaq sonra da semicolon qoymaq lazımdır. Compileri kind of warn edir ki, bele bir funksiya olacaq.

```
#include <stdio.h>

void meow(void);

int main(void)
{
    for (int i = 0; i < 3; i++)
    {
        meow();
    }
}

void meow(void)
{
    printf("meow\n");
}
```

Void means no output or input.

Variables only exist in the scope they exist, which is curly braces {}. Asağıdaki kodda buna gore de, variable'i birinci colde declare edenden sonra iceride input isteyirik. Bu kodun meqsedi odur ki, input "yanlis" verilse, terminal senden defelerle yeniden input istesin, duzgün input yazana qeder. Do-while kind of birlesikdi, weird gorsense de. "Do the following while" kimi deyilir. Inputla deal etmek ucuñ en recommendable yoldur bu. While dovrunden ferqli olaraq emin olmaq olur ki , burada expression en azi 1 defe icra olunacaq.

```
void meow(int n);

int main(void)
{
    int n;
    do
    {
        n = get_int("Number: ");
    }
    while (n < 1);
    meow(n);
}
```

```

}

void meow(int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("meow\n");
    }
}

```

Eger variablein hec deyismeyini istemirsense, calissan bele sistem redd elesin isteyirsense const isletmelisen.

```
1. const int n = 3
```

Comment - //

Integer overflow olmasi bir ededin yaddasdan cox tutmasidir. O zaman meselen 3bitlikdirse, 111den sonra 1000 gelir, hansiki en soldaki 1-l store elemeye yer yoxdur, ona gore de her sey 0lanmis olur. Bunun olmamasi ucin “long” istifade etmek olar int evezine. Hansiki 32 bit yox 64 bitlikdir.

Printf’in icinde variable istifade edende birdefe declare edib “%i\n” yazmagin besdir. Yeni, tekrar tekrar 2 defe iki variable ucin yazmaga ehtiyac yoxdur.

```

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    int x = get_int("x: ");

    int y = get_int("y: ");

    printf("%i\n", x + y);
}

```

C language’de intlerin cavabi float olabilmez, pythondan ferqli olaraq. Yeni 3/2 ni print edende “%i\n” edirsense, cavab integer olacaq, eger “%f\n” yazsan ise error olacaq, cunki durduq yere integerleri float kimi qebul et demeyin mumkun deyil. Instead asagidaki kod

kimi yaza bilersen. (float) makes sure that ondan sonraki seyi compiler float kimi qebul elesin.

```
#include <cs50.h>
#include <stdio.h>

int main(void)
{
    int x = get_int("x: ");

    int y = get_int("y: ");

    printf("%f\n", (float) x / y);
}
```

Printing Values

Placeholder=format code. Printf'in icinde % ile yazilan ifadelerdir

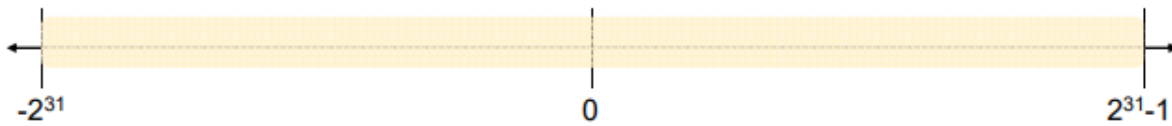
Types and format codes

Numbers	Text	True/False
int (%i)	char (%c)	bool (%i)
float (%f)	string (%s)	

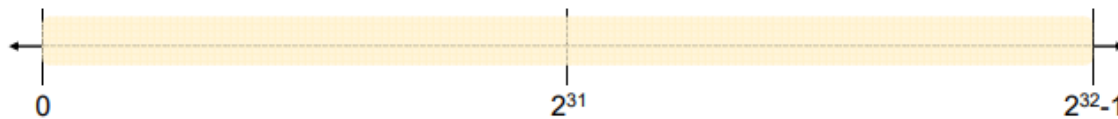
Data Types

unsigned int - unsigned is a *qualifier* that can be applied to certain types (including int), which effectively doubles the positive range of variables of that type, at the cost of disallowing any negative values. See the differences of the pictures below:

Integer Range



Unsigned Integer Range



double – float tiplerde, noqteden sonraki reqemlerin duzgunluyunun limiti vardir, overflow olur deye. O duzgunluk limitini artirmaq ucun double'dan istifade etmek mumkundur hansı ki, onda eded 32 bit yox 64 bit tutmus olur.

void – it is a type, but not a data type. Functions can have a void return type, which just means they don't return a value. You can't assign a variable void type, because it is not a data type.

Eyni anda bir nece variable'l declare etmek ucun bir defe type arasinda verguller sonra da identifier yazmaq bes edir. Asagida numune:

- If you wish to create multiple variables of the same type, you specify the type name *once*, and then list as many variables of that type as you want.

```
int height, width;  
float sqrt2, sqrt3, pi;
```

Operators

```
#include <cs50.h>  
#include <stdio.h>
```

```
int main(void)
{
    int num = -3;

    if (num)
    {
        printf("true\n");
    }
    else
    {
        printf("false\n");
    }
}

//prints true, because every nonzero number is true.
```

`||` - means or. `&&` - means and.

Two types of Boolean expressions:

- **Logical operator** – NOT, OR, AND
- **Relational operators** – comparators: <, >, >=, <=, ==, !=.

Conditionals

Switch() conditionali vardır ki, ya *integer* ya *char* tipinde olan birseye uygun conditionlar verir. Eger biri uygunlarsa ve break olursa hemin girintidekileri icra edir, break olunmasa ise, uyduğu ve ondan sonra gelen case'leri icra edir.

Syntax

```
switch (expression) {
    case x:
        // code block
        break;
    case y:
        // code block
        break;
    default:
        // code block
}
```

```

#include <cs50.h>
#include <stdio.h>

int main(void)
{
    char day = get_char("Which weekday? ");

    switch (day) {
        case 'm':
            printf("Monday");
            break;
        case 't':
            printf("Tuesday");
            break;
        case 'w':
            printf("Wednesday");
            break;
        case 'T':
            printf("Thursday");
            break;
        case 'f':
            printf("Friday");
            break;
        case 's':
            printf("Saturday");
            break;
        case 'S':
            printf("Sunday");
            break;
    }
}

```

Conditionalin yigcam yazilma usulu - **?:** ileidir.

```

int x = (expr) ? 5 : 6;

int x;
if (expr)
{
    x = 5;
}
else
{
    x = 6;
}

int x = (expr) ? 5 : 6;

```

Loops

For loopunda eger i 2 semicolon arasindaki boolean expression'u true elemirse dovrun govdesine hec girilmemis olacaq. Yeni error deyil.

```
1. for (int i = 0; i < 10; i++)  
2. {  
3. }
```

Loops

while

- Use when you want a loop to repeat an unknown number of times, and possibly not at all.

do-while

- Use when you want a loop to repeat an unknown number of times, but at least once.

for

- Use when you want a loop to repeat a discrete number of times, though you may not know the number at the moment the program is compiled.

Linux command-line

ls yazib liste baxanda, qara rengde gelenler txt/source code fayllaridir, mavi rengdekiler navigatable fayllardir, ve yasil rengde olanlar ise executabledir.

CTRL+L – clears the terminal.

cp ile fayllari kopyalamaq olursa da, directoryleri olmur onuncun, **cp -r** isletmek lazimdir, hansi ki, recursive copy demekdir, yeni directorynin icindeki hersey ile birlikde copy edir.

rm istifade ederek silende, testiqlemek ucun y-n sorusur, amma eger yuz faiz silinecekse ve tesdiq istemirsense **rm -f** yeni forcelamaqdan istifade ede bilersen.

Commandlardan “mv” commandi, hem file’l move edir hem de elebir file adi yoxdursa, filein adini yazilan ada deyisir.

Magic Numbers

Meselen bir konstantdan istifade etmeli olsan, pi kimi, 3.14159 kimi uzun ededi tekrar ve tekrar yazmaq istemesen, onu bir constanta menimsede bilersen, amma bu konstantdir deye zaten deyismeyecek, ona gore de global sekilde tanitmaq lazim olur. O zaman **#define** istifade olunur. **#define NAME REPLACEMENT**. Constant identifierin adi uppercase ile yazilir ki, variable’lardan secilsin. NAME identifierini compiler gorende, REPLACEMENT ile evez eletdirir. Meselen asagidaki pseudocode’daki yazilisa bax.

```
#define DECKSIZE 52
card deal_cards(deck name)
{
    for (int i = 0; i < DECKSIZE; i++)
    {
        // deal the card
    }
}
```

Better than

```
card deal_cards(deck name)
{
    int deck_size = 52;
    for (int i = 0; i < deck_size; i++)
    {
        // deal the card
    }
}
```