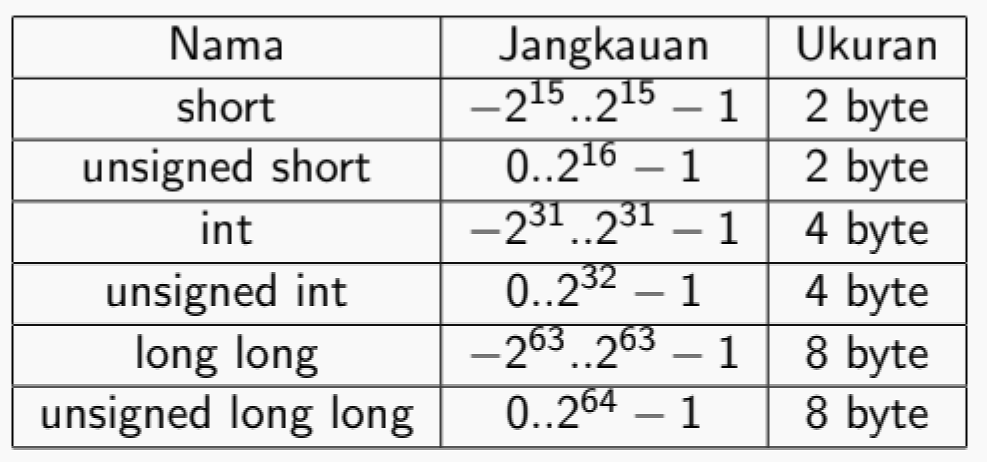
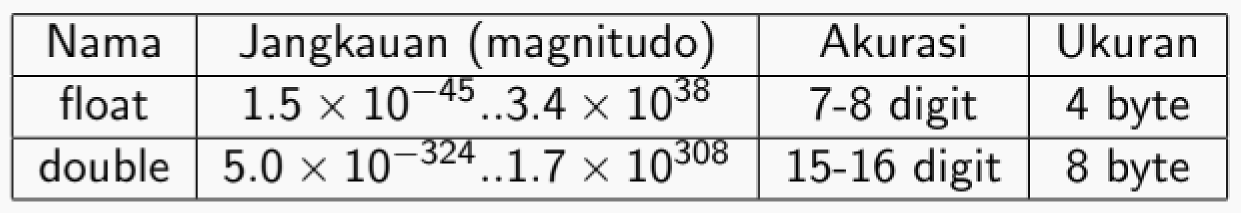
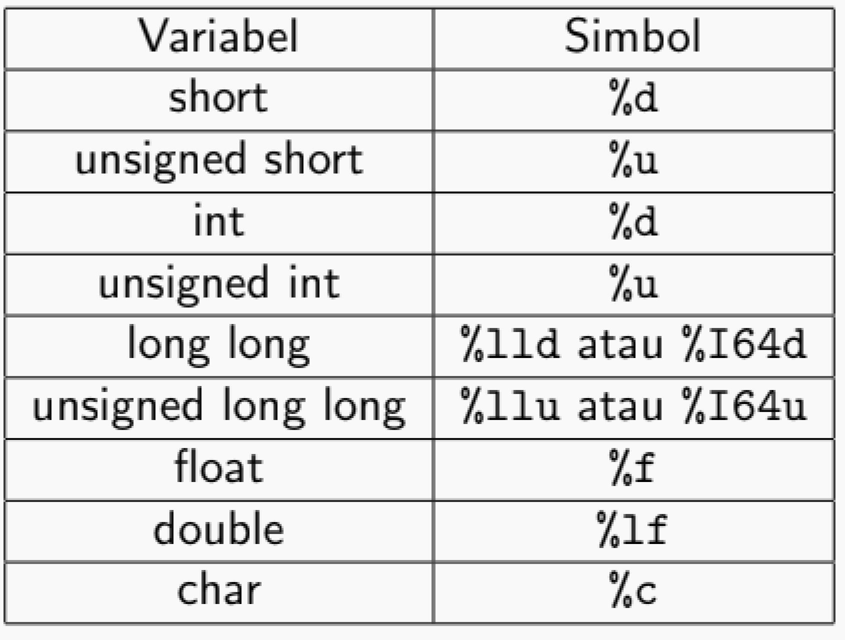


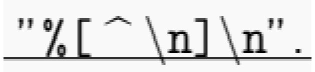
Nama	Jangkauan	Ukuran
short	$-2^{15}..2^{15}-1$	2 byte
unsigned short	$0..2^{16}-1$	2 byte
int	$-2^{31}..2^{31}-1$	4 byte
unsigned int	$0..2^{32}-1$	4 byte
long long	$-2^{63}..2^{63}-1$	8 byte
unsigned long long	$0..2^{64}-1$	8 byte



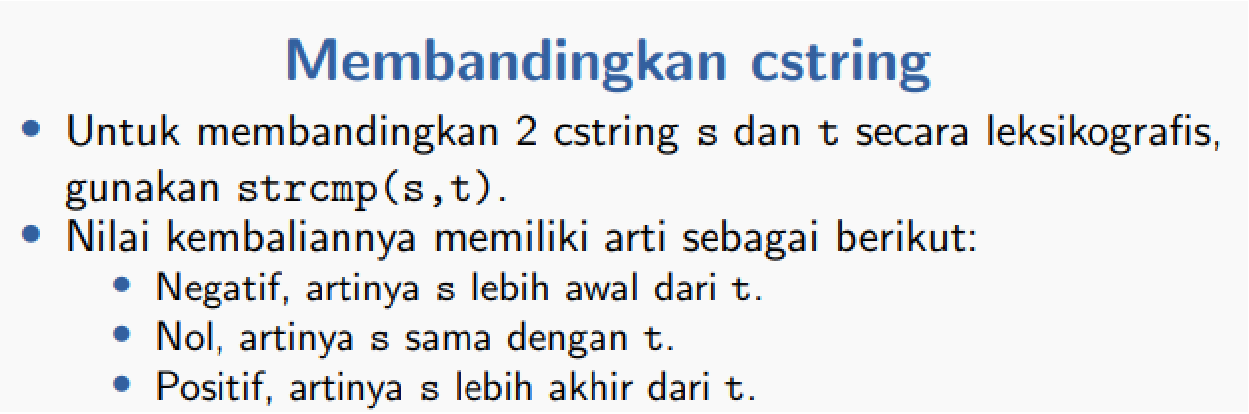
Nama	Jangkauan (magnitudo)	Akurasi	Ukuran
float	$1.5 \times 10^{-45}..3.4 \times 10^{38}$	7-8 digit	4 byte
double	$5.0 \times 10^{-324}..1.7 \times 10^{308}$	15-16 digit	8 byte



Variabel	Simbol
short	`%d`
unsigned short	`%u`
int	`%d`
unsigned int	`%u`
long long	`%lld` atau `%I64d`
unsigned long long	`%llu` atau `%I64u`
float	`%f`
double	`%lf`
char	`%c`

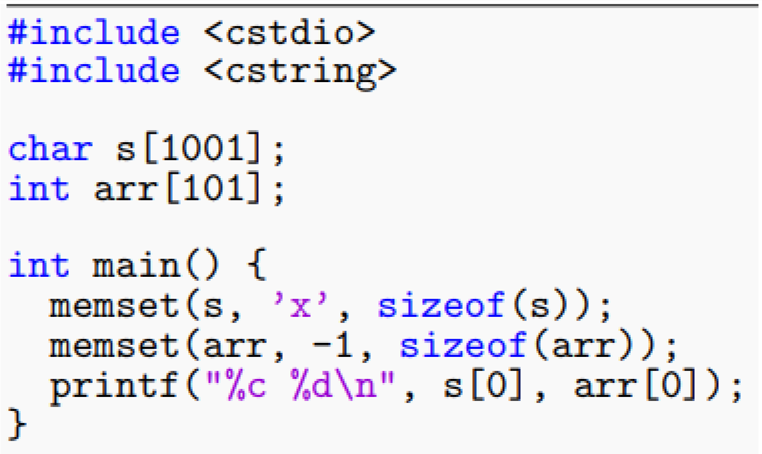


```
"%[^\n]\n".
```

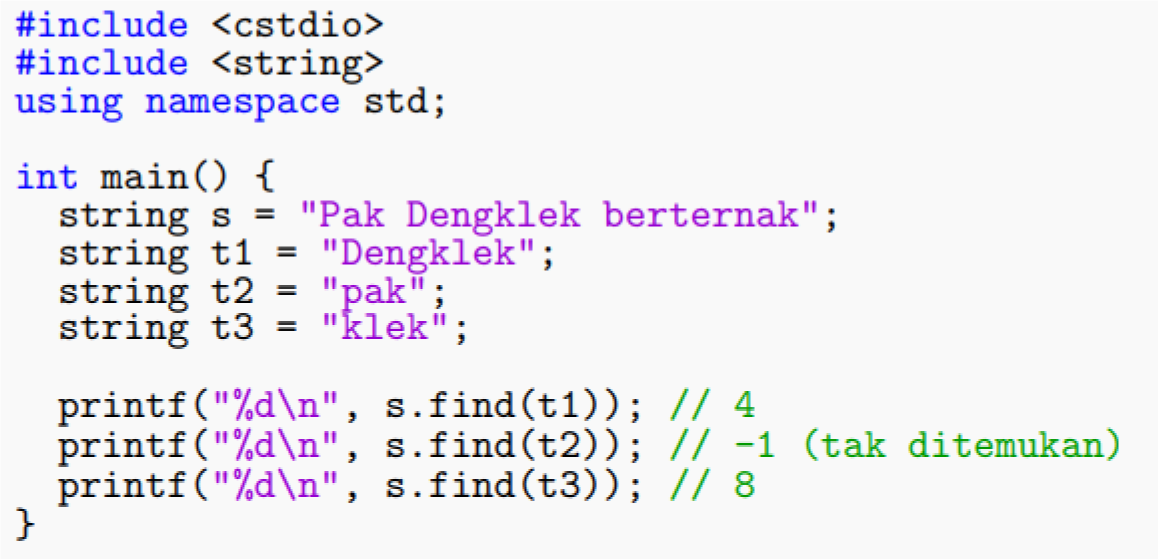


Membandingkan cstring

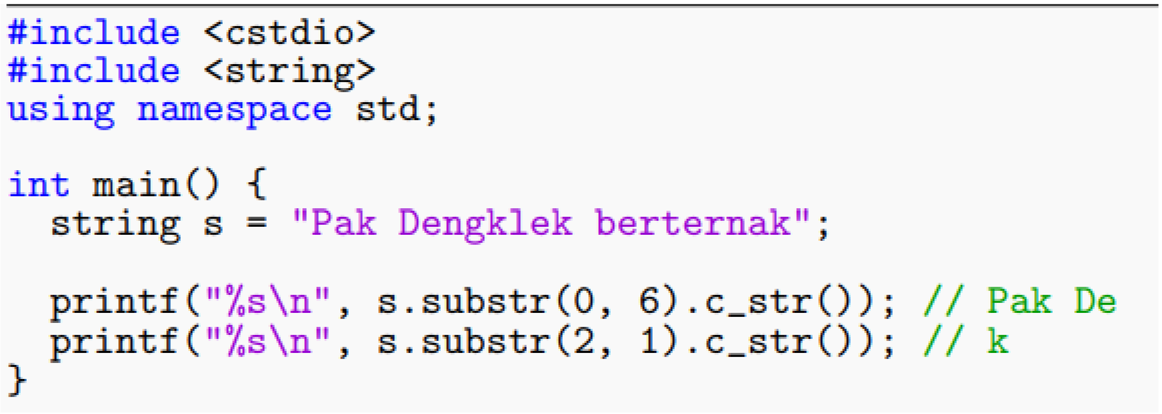
- Untuk membandingkan 2 cstring `s` dan `t` secara leksikografis, gunakan `strcmp(s,t)`.
- Nilai kembaliannya memiliki arti sebagai berikut:
  - Negatif, artinya `s` lebih awal dari `t`.
  - Nol, artinya `s` sama dengan `t`.
  - Positif, artinya `s` lebih akhir dari `t`.



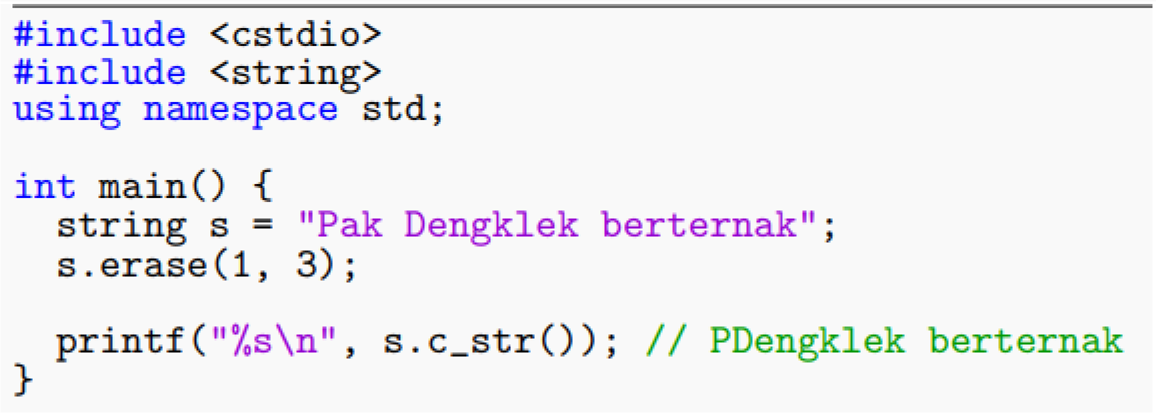
```
#include <cstdio>
#include <cstring>

char s[1001];
int arr[101];

int main() {
  memset(s, ’x’, sizeof(s));
  memset(arr, -1, sizeof(arr));
  printf("%c %d\n", s[0], arr[0]);
}
```



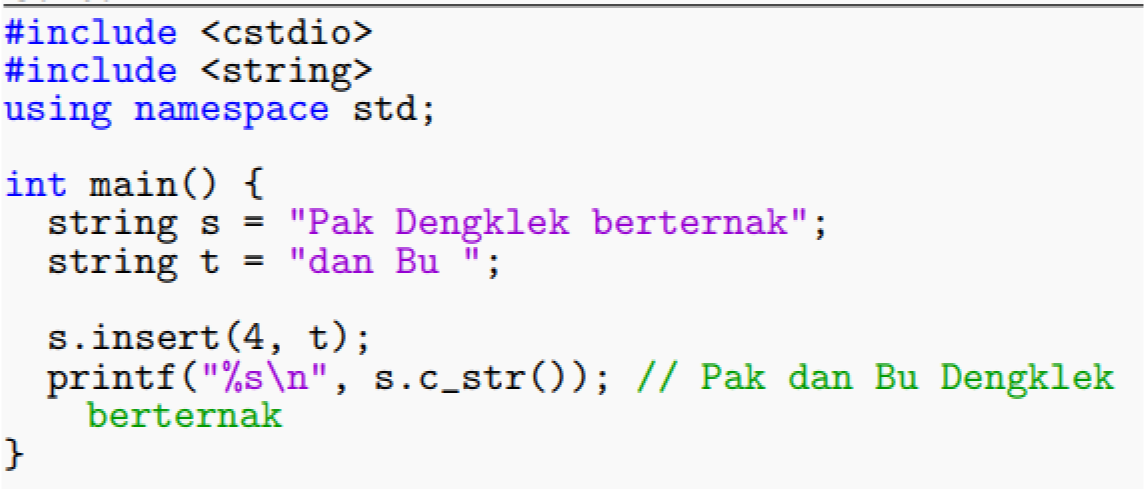
```
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "Pak Dengklek berternak";
  string t1 = "Dengklek";
  string t2 = "pak";
  string t3 = "klek";

  printf("%d\n", s.find(t1)); // 4
  printf("%d\n", s.find(t2)); // -1 (tak ditemukan)
  printf("%d\n", s.find(t3)); // 8
}
```



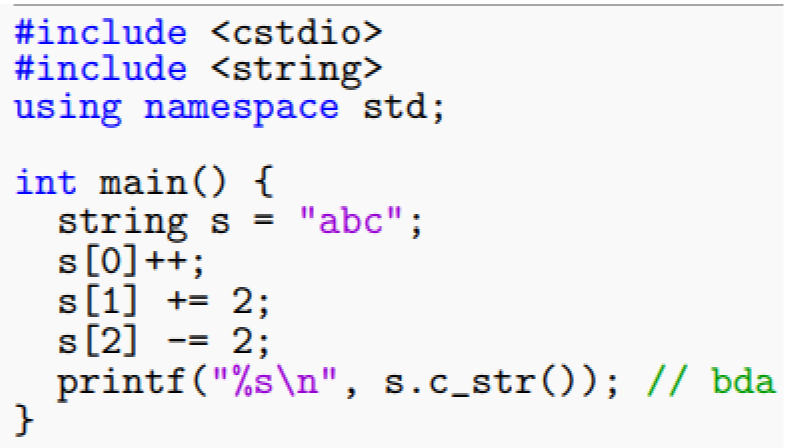
```cpp
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "Pak Dengklek berternak";

  printf("%s\n", s.substr(0, 6).c_str()); // Pak De
  printf("%s\n", s.substr(2, 1).c_str()); // k
}
```



```cpp
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "Pak Dengklek berternak";
  s.erase(1, 3);

  printf("%s\n", s.c_str()); // PDengklek berternak
}
```



```cpp
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "Pak Dengklek berternak";
  string t = "dan Bu ";

  s.insert(4, t);
  printf("%s\n", s.c_str()); // Pak dan Bu Dengklek
    berternak
}
```

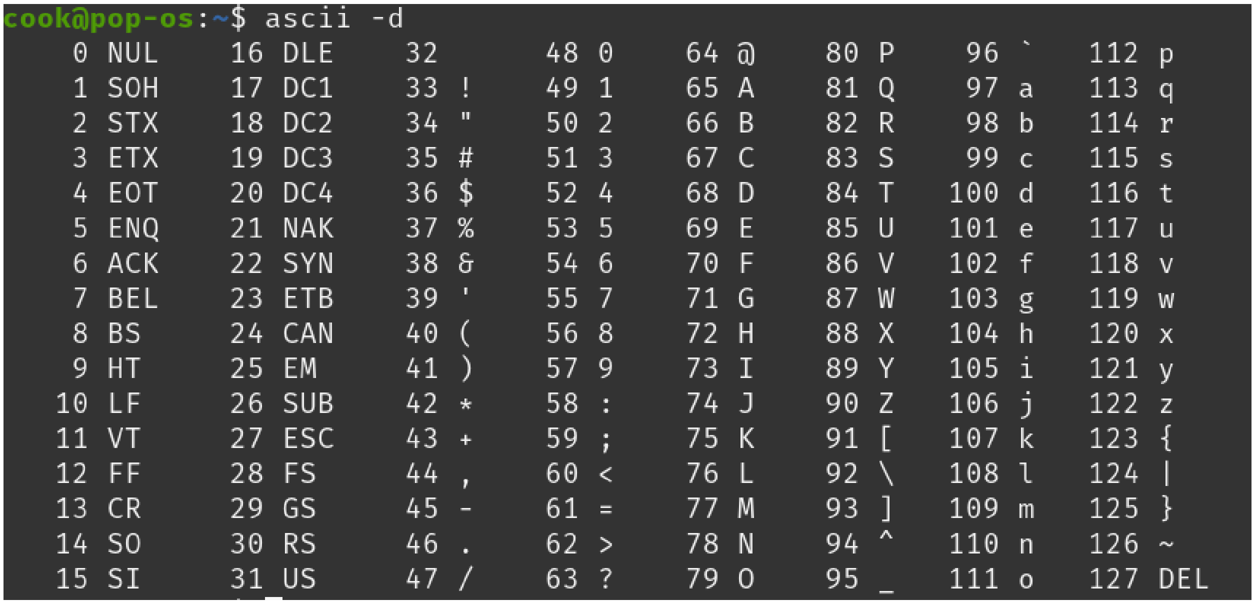


```cpp
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "abc";
  s[0]++;
  s[1] += 2;
  s[2] -= 2;
  printf("%s\n", s.c_str()); // bda
}
```

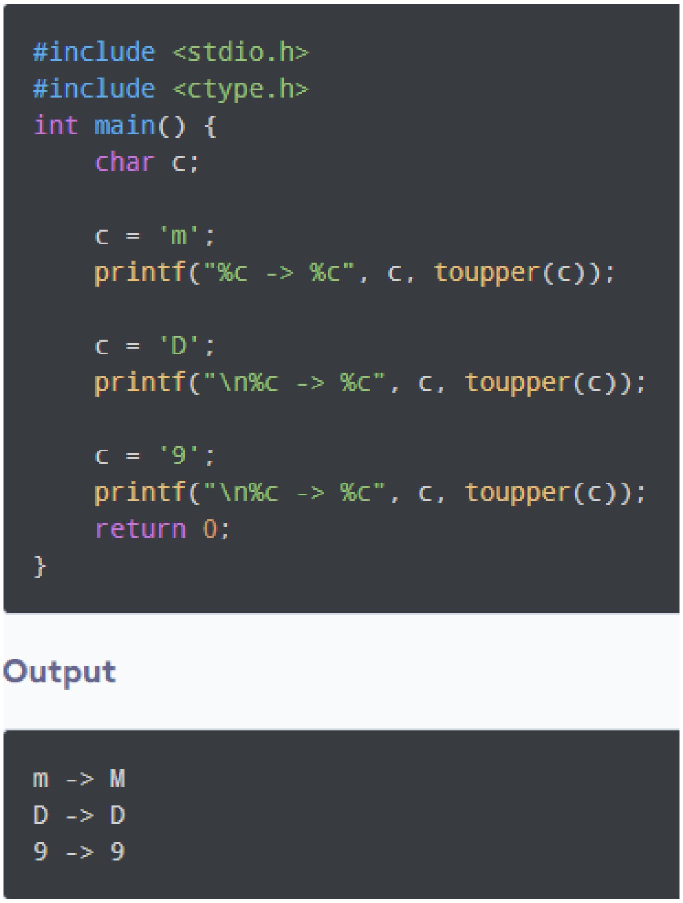
Compile :

“g++ file.java -o output.exe” atau “g++ file.java”

“./output.exe” atau “a.exe”



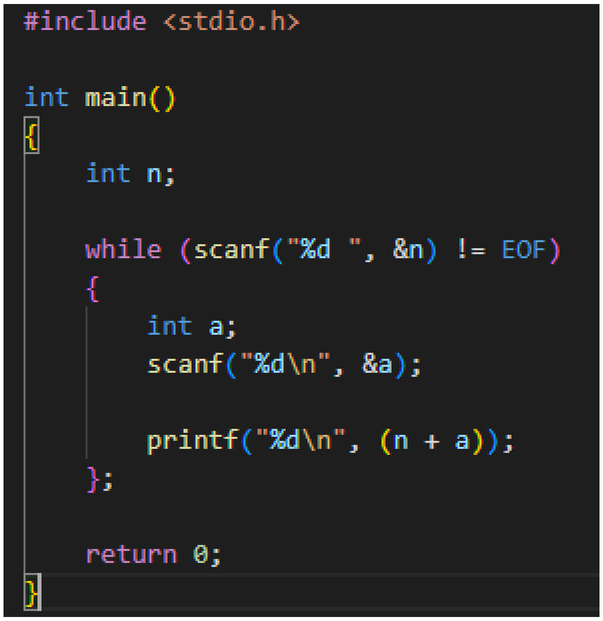
```
cook@pop-os:~$ ascii -d
    0 NUL    16 DLE    32        48 0    64 @    80 P     96 `    112 p
    1 SOH    17 DC1    33 !      49 1    65 A    81 Q     97 a    113 q
    2 STX    18 DC2    34 "      50 2    66 B    82 R     98 b    114 r
    3 ETX    19 DC3    35 #      51 3    67 C    83 S     99 c    115 s
    4 EOT    20 DC4    36 $      52 4    68 D    84 T    100 d    116 t
    5 ENQ    21 NAK    37 %      53 5    69 E    85 U    101 e    117 u
    6 ACK    22 SYN    38 &      54 6    70 F    86 V    102 f    118 v
    7 BEL    23 ETB    39 '      55 7    71 G    87 W    103 g    119 w
    8 BS     24 CAN    40 (      56 8    72 H    88 X    104 h    120 x
    9 HT     25 EM     41 )      57 9    73 I    89 Y    105 i    121 y
   10 LF     26 SUB    42 *      58 :    74 J    90 Z    106 j    122 z
   11 VT     27 ESC    43 +      59 ;    75 K    91 [    107 k    123 {
   12 FF     28 FS     44 ,      60 <    76 L    92 \    108 l    124 |
   13 CR     29 GS     45 -      61 =    77 M    93 ]    109 m    125 }
   14 SO     30 RS     46 .      62 >    78 N    94 ^    110 n    126 ~
   15 SI     31 US     47 /      63 ?    79 O    95 _    111 o    127 DEL
```



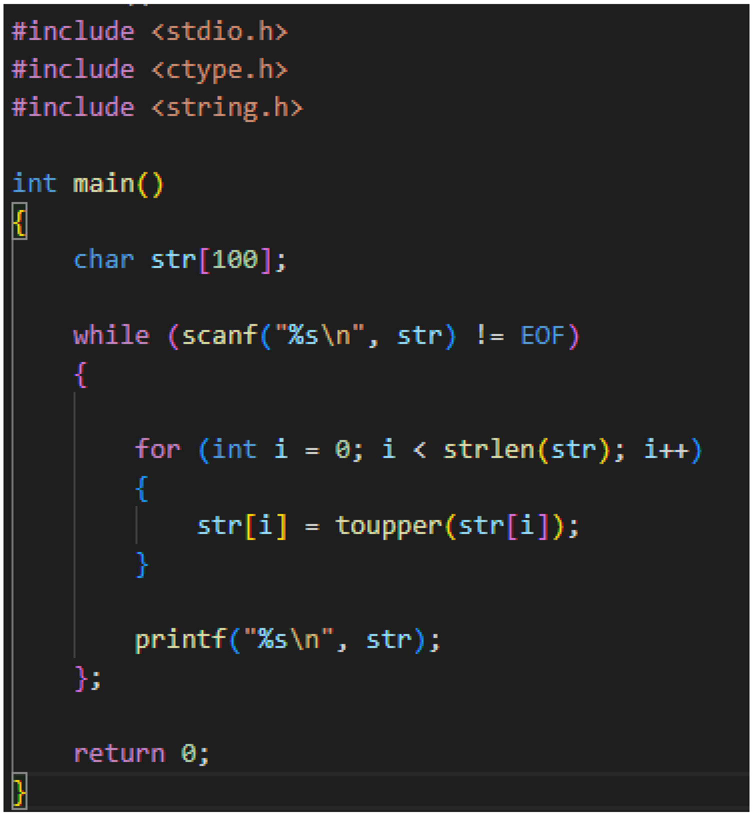
```c
#include <stdio.h>
#include <ctype.h>
int main() {
    char c;

    c = 'm';
    printf("%c -> %c", c, toupper(c));

    c = 'D';
    printf("\n%c -> %c", c, toupper(c));

    c = '9';
    printf("\n%c -> %c", c, toupper(c));
    return 0;
}
```

Output

```
m -> M
D -> D
9 -> 9
```



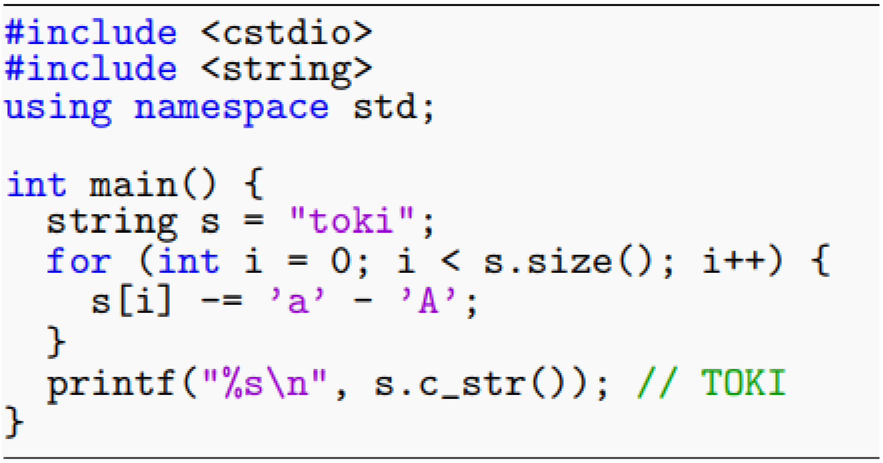
```c
#include <stdio.h>

int main()
{
    int n;

    while (scanf("%d ", &n) != EOF)
    {
        int a;
        scanf("%d\n", &a);

        printf("%d\n", (n + a));
    };

    return 0;
}
```



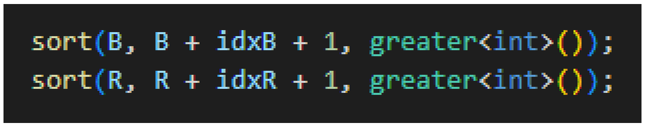
```
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int main()
{
    char str[100];

    while (scanf("%s\n", str) != EOF)
    {

        for (int i = 0; i < strlen(str); i++)
        {
            str[i] = toupper(str[i]);
        }

        printf("%s\n", str);
    };

    return 0;
}
```

// toupper -> toUpperCase

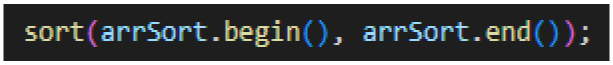
Cara lain:



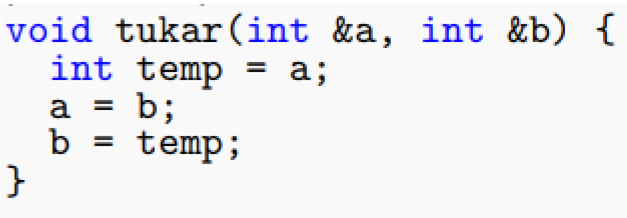
```
#include <cstdio>
#include <string>
using namespace std;

int main() {
  string s = "toki";
  for (int i = 0; i < s.size(); i++) {
    s[i] -= 'a' - 'A';
  }
  printf("%s\n", s.c_str()); // TOKI
}
```



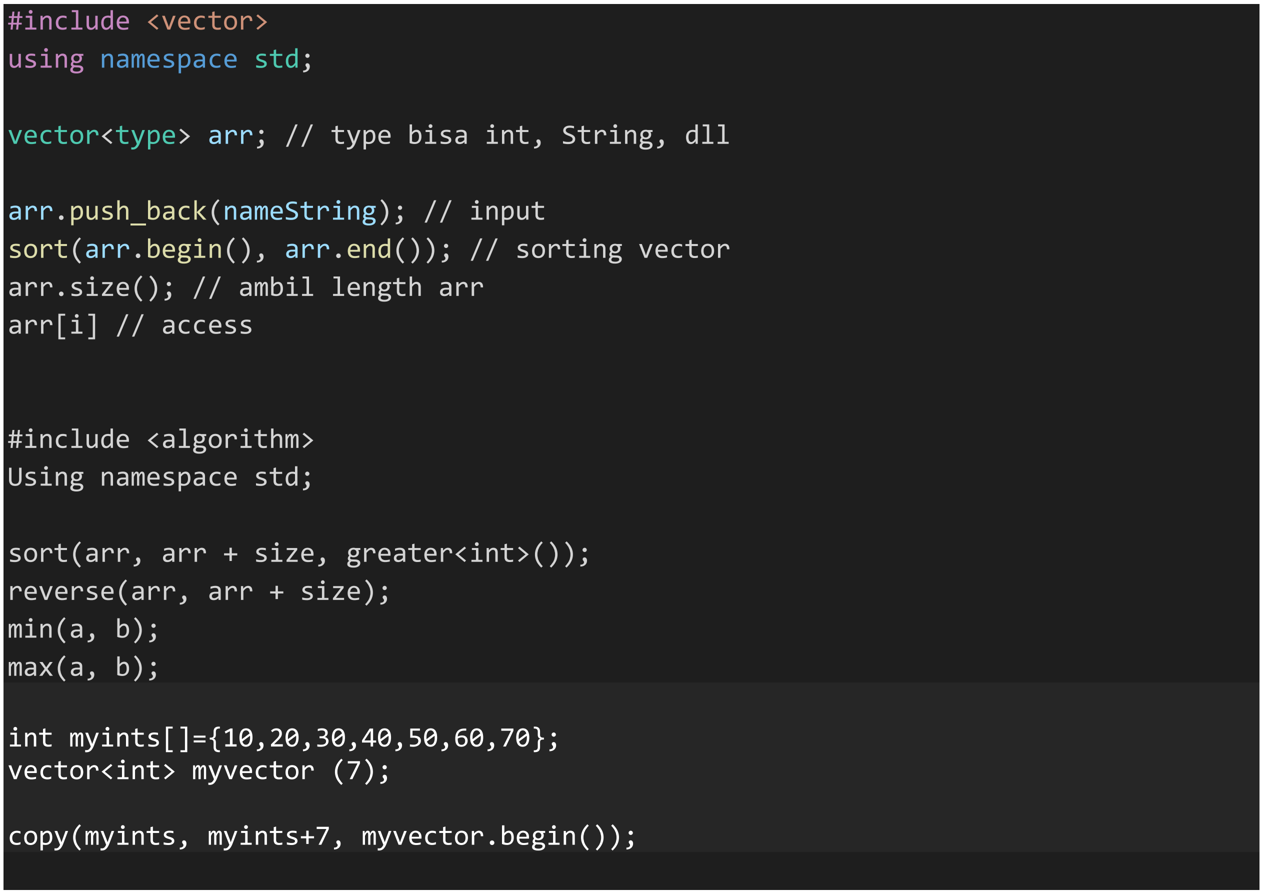
```
sort(B, B + idxB + 1, greater<int>());
sort(R, R + idxR + 1, greater<int>());
```



```
sort(arrSort.begin(), arrSort.end());
```



```
void tukar(int &a, int &b) {
  int temp = a;
  a = b;
  b = temp;
}
```



```
#include <vector>
using namespace std;

vector<type> arr; // type bisa int, String, dll

arr.push_back(nameString); // input
sort(arr.begin(), arr.end()); // sorting vector
arr.size(); // ambil length arr
arr[i] // access


#include <algorithm>
Using namespace std;

sort(arr, arr + size, greater<int>());
reverse(arr, arr + size);
min(a, b);
max(a, b);

int myints[]={10,20,30,40,50,60,70};
vector<int> myvector (7);

copy(myints, myints+7, myvector.begin());
```