

## Функција rand()

Функција `rand()` одређује низ псеудослучајних целих бројева у опсегу од 0 до `RAND_MAX`, који је дефинисан у библиотеци `<stdlib.h>`.

Псеудослучајни значи да избор целих бројева није заиста случајан, већ је предвиђен алгоритмом који се стартује позивом функције.

Пример:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int x;
    printf("Ispisuje 100 pseudoslucajnih brojeva!\n");
    for(x=0;x<100;x++)
        printf("%d\t",rand());
    return(0);
}
```

После два стартовања кода, на екрану се исписује два пута идентичан низ 100 бројева:

Ispisuje 100 pseudoslucajnih brojeva!

```
41  18467 6334 26500 19169 15724 11478 29358 26962 24464
5705 28145 23281 16827 9961 491 2995 11942 4827 5436
32391 14604 3902 153 292 12382 17421 18716 19718 19895
5447 21726 14771 11538 1869 19912 25667 26299 17035 9894
28703 23811 31322 30333 17673 4664 15141 7711 28253 6868
25547 27644 32662 32757 20037 12859 8723 9741 27529 778
12316 3035 22190 1842 288 30106 9040 8942 19264 22648
27446 23805 15890 6729 24370 15350 15006 31101 24393 3548
19629 12623 24084 19954 18756 11840 4966 7376 13931 26308
16944 32439 24626 11323 5537 21538 16118 2082 22929 16541
```

Проблем псеудослучајних бројева се може донекле решити коришћењем функције `srand((unsigned)())`. Сада почетна вредност за генерисање псеудослучајних бројева зависи од целобројне вредности у загради функције `srand((unsigned)())`.

Ако би корисник могао по нахођењу да уноси такву почетну вредност, као излаз функције `rand()` би се добијали случајни бројеви мање зависни од алгоритма за генерисање случајних вредности.

Пример:

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int x;
```

```

char s[6];
printf("Unesi broj za sidovanje (2 - 65000) : ");
srand((unsigned)atoi(gets(s)));
puts("Ispisuje 100 slucajnih brojeva u zavisnosti od sidera!\n");
for(x=0;x<100;x++)
    printf("%d\t",rand());
return(0);
}

```

Сваки различити број за сидовање омогућава различите случајне бројеве на излазу:

Unesi broj za sidovanje (2 - 65000) : 2345

Ispisuje 100 slucajnih brojeva u zavisnosti od sidera!

```

7696 14140 2448 4411 13748 12170 32567 16338 31121 31313
19461 318 12294 4103 25574 8656 15346 12443 3425 31487
18879 27746 5824 25601 19872 14676 26226 4248 14102 26744
27309 21039 32185 3997 11352 28453 27445 15179 2343 1129
18067 31993 22792 9543 17594 10624 702 11649 25141 23504
17355 31138 2614 4066 12180 31119 11995 27118 10688 20831
14001 32221 24545 31852 25463 11009 1622 9898 5668 24580
23664 12344 5163 23326 15315 3893 12797 14987 14111 981
1743 28989 3024 21172 3486 29646 27491 31082 22385 12728
24507 4447 13023 11976 20162 21729 27856 20688 24915 31347

```

Unesi broj za sidovanje (2 - 65000) : 10000

Ispisuje 100 slucajnih brojeva u zavisnosti od sidera!

```

32694 12743 11816 20150 935 23032 3275 21360 8401 22703
1351 9815 9291 17910 4497 21341 15564 24059 25646 19910
12315 12470 26949 15590 1222 29900 24056 1580 29719 435
29172 14431 14404 13591 10055 15400 2790 18301 25715 28113
26540 21543 27840 1100 26449 6199 13014 26999 18123 11543
14730 26526 28358 16919 30457 20545 3920 6488 28115 24
9174 19434 28922 25685 29519 6449 17740 25897 24759 17805
5564 22921 8127 19513 28072 26674 8643 26740 11623 31211
20220 14815 11553 8458 24567 2087 483 6713 27461 27107
6386 19066 31605 21184 7520 24169 16703 19675 23518 28353

```

Проблем је што ово захтева да се у програму тражи унос од корисника сваки пут пре него што би се генерисао случајан број. Значи да би било боље пронаћи механизам самосталног генерисања (сидовања) случајних бројева.

Један од начина аутоматског генерисања различитих вредности у сваком тренутку рада процесора јесте функција `time(NULL)` која даје као резултат време протекло од неког почетног тренутка од којег се мери преоцесорско време. Функција се налази у библиотеци `<time.h>` која декларише типове и функције за операције са датумима и временом.

Пример:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{
    int x;
    srand((unsigned)time(NULL));
    puts("Ispisuje 100 slucajnih brojeva u zavisnosti od tajmera!\n");
    for(x=0;x<100;x++)
        printf("%d\t",rand());
    return(0);
}
```

Ispisuje 100 slucajnih brojeva u zavisnosti od tajmera!

```
21773 29166 11808 19126 9312 24654 3122 31279 26920 1950
26506 23949 8994 24756 11643 25953 5867 6858 6436 17895
14112 28698 1449 4152 10436 10411 13764 27584 26338 16234
13437 1011 18803 23932 13109 32316 29569 8147 2281 3126
27789 23295 14005 28191 26927 20807 5527 31114 1788 3833
4470 30803 32684 24830 5475 29656 6013 6577 3358 16486
19154 6130 8156 23794 32544 17525 2119 31498 15114 31072
11719 17163 10699 2070 25344 10387 28117 20813 28060 25401
1719 9063 15178 24444 20036 21247 26630 15354 20722 16834
31132 9955 1408 18304 29799 12204 10204 12674 11615 24840
```

Сада свако следеће стартовање кода даје другачије случајне бројеве.

Једини проблем је што добијени случајни бројеви имају четири или пет цифара што често није корисно за потребе корисника. Да би се смањио број потребних цифара случајних бројева користи се функција `rand()%10`. Сада је ограничен излаз функције на целе позитивне бројеве, од 0 до 9.

Пример:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{
    int x;
    srand((unsigned)time(NULL));
    for(x=0;x<100;x++)
        printf("%i\t",rand()%10);
    return(0);
}
```

5	1	3	0	8	4	0	8	1	6
4	9	1	3	8	4	7	6	3	6
4	6	5	2	6	8	6	4	1	5
6	8	1	8	1	0	1	9	4	2
1	7	7	7	3	3	5	4	2	3
5	1	8	7	6	3	6	3	9	6
3	9	7	5	7	0	0	8	8	2
8	8	0	5	2	1	7	2	5	0
6	1	0	4	9	3	5	5	3	6
8	2	0	9	0	0	0	8	2	3

-----

7	2	4	9	0	0	2	7	2	4
9	5	8	1	7	7	7	2	9	5
7	6	2	9	6	2	1	0	0	2
1	7	9	8	7	1	3	2	2	6
0	2	7	5	4	9	5	8	6	8
5	9	9	3	8	6	5	5	2	5
6	9	0	9	2	4	1	1	5	0
7	7	9	2	4	9	3	4	4	9
9	3	0	9	9	5	5	6	0	4
5	4	7	7	0	2	2	3	3	7

На овај начин се генеришу случајни бројеви у опсегу од 0-9. Случајни бројеви нису апсолутно случајни али је максимално смањен утицај предодређеног алгоритма за њихово генерисање. Овај поступак омогућава коришћење представљеног кода за потребе програмирања видео игара.