

1. Введение

- 1) Текстовая формулировка задачи
- 2) Пример кода, решающего данную задачу
- 3) Скриншот программы

2. Ход работы

2.1. Код приложения

Генератор случайных чисел Парка-Миллера с перетасовкой

```
#include <stdio.h>
#define IA 16807
#define IM 2147483647
#define AM (1./IM)
#define IQ 12773
#define IR 2836
#define NTAB 32
#define NWUP 8
#define NDIV (1+(IM-1)/NTAB)
#define EPS 1.2e-7
#define RNMX (1.0-EPS)
#define MASK 123456789

static long dummy;
void Seed(long dum) {
    dummy = dum;
}

float unirand0(void) {
    long k;
    float ans;

    dummy ^= MASK;
    k = dummy / IQ;

    if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
        dummy += IM;

    ans = AM * dummy;

    dummy ^= MASK;
```

```

return(ans);
}

float unirand1(void) {
int j;
long k;
static long iy = 0, iv[NTAB];
float temp;

if (dummy <= 0 || !iy) {
if (dummy < 0) dummy = -dummy; else
if (dummy == 0) dummy = 1;
for (j = NTAB + NWUP - 1; j >= 0; j--) {
k = dummy / IQ;

if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

if (j < NTAB)
iv[j] = dummy;
}

iy = iv[0];
}

k = dummy / IQ;
if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
dummy += IM;

iy = iv[j = iy / NDIV];
iv[j] = dummy;

if ((temp = AM * iy) > RNMx)
return(RNMx);
else
return(temp);
}

int main() {
int i;
Seed(6723);
for (i = 0; i < 100; i++)
printf("%f\n", unirand1());
}

```

```
}
```

Генератор случайных чисел Парка-Миллера без перетасовки

```
#include <stdio.h>
#define IA 16807
#define IM 2147483647
#define AM (1./IM)
#define IQ 12773
#define IR 2836
#define MASK 123456789

static long dummy;
void Seed(long dum) {
    dummy = dum;
}

float unirand0(void) {
    long k;
    float ans;

    dummy ^= MASK;
    k = dummy / IQ;

    if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
        dummy += IM;

    ans = AM * dummy;

    dummy ^= MASK;

    return(ans);
}

int main() {
    int i;
    Seed(6723);
    for (i = 0; i < 100; i++)
        printf("%f\n", unirand0());
}
```

2.2. формулы

Общая формула генератора случайных чисел $X_{k+1} = a * X_k \bmod m$

3. Пример скриншота программы

Генератор случайных чисел Парка-Миллера с перетасовкой

```
1 #include <stdio.h>
2 #define IA 16807
3 #define IM 2147483647
4 #define AM (1./IM)
5 #define IQ 12773
6 #define IR 2836
7 #define NTAB 32
8 #define NMUP 8
9 #define NDIV (1+(IM-1)/NTAB)
10 #define EPS 1.2e-7
11 #define RNMN (1.0-EPS)
12 #define MASK 123456789
13 static long dummy;
14 void Seed(long dum) {
15     dummy = dum;
16 }
17 float unrand0(void) {
18     long k;
19     float ans;
20
21     dummy ^= MASK;
22     k = dummy / IQ;
23
24     if ((dummy = IA + (dummy - k * IQ) - IR * k) < 0)
25         dummy += IM;
26     ans = AM + dummy;
27     dummy ^= MASK;
28 }
29
30
```

0.872814
0.841861
0.981817
0.843414
0.848921
0.896237
0.855248
0.835611
0.882865
0.879655
0.882892
0.838688
0.885687
0.866585
0.866464
0.824135
0.821179
0.865644
0.837348
0.986494
0.847323
0.829146
0.885818
0.892465
0.863126
0.875276
0.862347
0.864648
0.833595
0.837656
0.841818
0.885193
0.863112

Рис. 1. скриншот программы

```
21 dummy ^= MASK;
22 k = dummy / IQ;
23
24 if ((dummy = IA + (dummy - k * IQ) - IR * k) < 0)
25     dummy += IM;
26
27 ans = AM * dummy;
28
29 dummy ^= MASK;
30
31 return(ans);
32 }
33 float unrand1(void) {
34     int j;
35     long k;
36     static long iy = 0, iv[NTAB];
37     float teap;
38
39     if (dummy <= 0 || !iy) {
40         if (dummy < 0) dummy = -dummy; else
41             if (dummy == 0) dummy = 1;
42         for (j = NTAB + NWUP - 1; j >= 0; j--) {
43             k = dummy / IQ;
44
45             if ((dummy = IA + (dummy - k * IQ) - IR * k) < 0)
46                 dummy += IM;
47
48             if (j < NTAB)
49                 iv[j] = dummy;
50         }
51     }
52     teap = dummy * teap + 0.5;
53     return(teap);
54 }
```

Рис. 2. скриншот программы

```
39 v if (dummy <= 0 || !iy) {
40     if (dummy < 0) dummy = -dummy; else
41         if (dummy == 0) dummy = 1;
42 v for (j = NTAB + NwUP - 1; j >= 0; j--) {
43     k = dummy / IQ;
44
45     if ((dummy = IA + (dummy - k + IQ) - IR + k) < 0)
46         dummy += IM;
47
48     if (j < NTAB)
49         iv[j] = dummy;
50 }
51 iy = iv[0];
52 }
53 k = dummy / IQ;
54 if ((dummy = IA + (dummy - k + IQ) - IR + k) < 0)
55     dummy += IM;
56
57 iy = iv[j] = iy / NDIV;
58 iv[j] = dummy;
59
60 if ((temp = AM + iy) > RMX)
61     return(RMX);
62 else
63     return(temp);
64 }
65 v int main() {
66     int i;
67     Seed(6723);
68     for (i = 0; i < 100; i++)
```

```
0.872814
0.841861
0.991817
0.843414
0.848921
0.896237
0.855248
0.835611
0.882865
0.879655
0.882892
0.838688
0.885687
0.866585
0.866464
0.824135
0.821178
0.865644
0.837248
0.996484
0.847323
0.829146
0.885818
0.892465
0.863126
0.875276
0.862347
0.864648
0.833595
0.837656
0.841818
0.885193
0.863112
>
```

Рис. 3. скриньшот программы

```
42 v for (j = NTAB + NWUP - 1; j >= 0; j--) {
43     k = dummy / IQ;
44
45     if ((dummy = IA + (dummy - k * IQ) - IR + k) < 0)
46         dummy += IM;
47
48     if (j < NTAB)
49         iv[j] = dummy;
50 }
51 iy = iv[0];
52 }
53 k = dummy / IQ;
54 if ((dummy = IA + (dummy - k * IQ) - IR + k) < 0)
55     dummy += IM;
56
57 iy = iv[j = iy / NDIV];
58 iv[j] = dummy;
59
60 if ((temp = AM + iy) > RNDX)
61     return(RNDX);
62 else
63     return(temp);
64 }
65 int main() {
66     int i;
67     Seed(6723);
68     for (i = 0; i < 100; i++)
69         printf("%f\n", unizand1());
70 }
71
```

0.872814
0.841861
0.981817
0.843414
0.848921
0.396237
0.855348
0.835611
0.882865
0.879655
0.882892
0.838688
0.885687
0.866585
0.866464
0.824135
0.821178
0.865644
0.837248
0.986484
0.847323
0.829146
0.885818
0.892465
0.863126
0.875276
0.862347
0.864648
0.833595
0.837656
0.841818
0.885193
0.863112
>

Рис. 4. скриншот программы

Генератор случайных чисел Парка-Миллера без перетасовкой

```
1 #include <stdio.h>
2 #define IA 16807
3 #define IN 2147483647
4 #define AM (1./IN)
5 #define IQ 12773
6 #define IR 2836
7 #define MASK 123456789
8 static long dummy;
9 void Seed(long dum) {
10     dummy = dum;
11 }
12 float unrand0(void) {
13     long k;
14     float ans;
15     dummy ^= MASK;
16     k = dummy / IQ;
17     if ((dummy = IA * (dummy - k * IQ) - IR * k) < 0)
18         dummy += IM;
19     ans = AM * dummy;
20     dummy ^= MASK;
21     return(ans);
22 }
23 int main() {
24     int i;
25     Seed(6723);
26     for (i = 0; i < 100; i++)
27         printf("%i\n", unrand0());
28 }
```

0.856678
0.859427
0.873718
0.813742
0.902880
0.825537
0.856371
0.835862
0.845654
0.818132
0.912824
0.873585
0.871461
0.896725
0.844794
0.854846
0.855967
0.811428
0.869413
0.862109
0.878211
0.833893
0.899955
0.833525
0.906984
0.853276
0.859540
0.884182
0.825538
0.822790
0.912411
0.874672
0.881157

Рис. 5. скриншот программы

4. библиографические ссылки

Для изучения «внутренностей» \TeX необходимо изучить [1], а для использования \LaTeX лучше почитать [2, 3].

Список литературы

- [1] Кнут Д.Э. Всё про \TeX . — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе \LaTeX . — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. \LaTeX в примерах. 2005 г.