

UYGULAMALAR

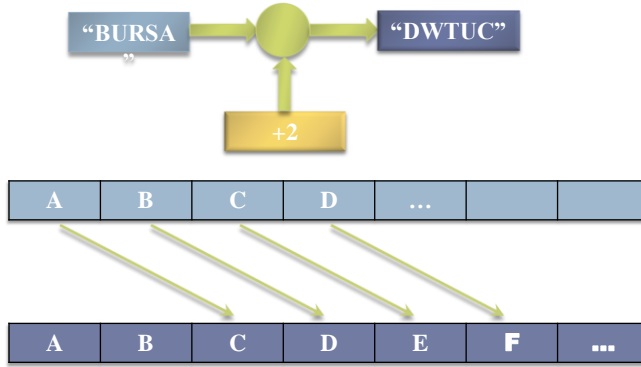
7. Hafta

Sezar Şifreleme Algoritması

İlk şifreleme algoritmalarından kabul edilen Sezar şifreleme algoritması (*Caesar chiper*), eski Roma İmparatoru *Julius Caesar* tarafından savaş zamanlarındaki bilgi gönderiminde kullanılmıştır. Bu algorithmada; mesajdaki her karakter, başka ('anahtar' değeri kadar ötelenmiş) karakterle yer değiştirerek şifreli mesaj elde edilmektedir. Örneğin ROT13 olarak adlandırılan şifreleme yönteminde öteleme miktarı 13'tür.

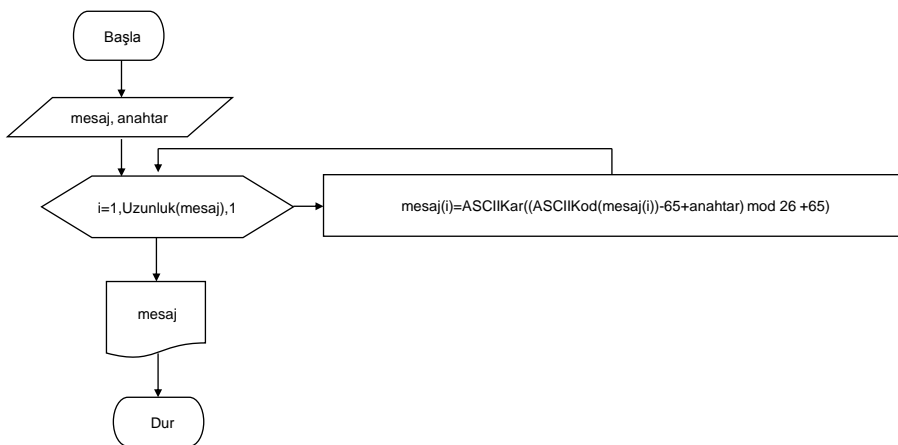


Sezar Şifreleme Algoritması



Eğer anahtar değeri 2 ise orijinal mesajdaki her harf, kendisinden iki sonraki harfle yer değiştirir. Yani orijinal mesajdaki "A" → "C", "B" → "D" olur.

Sezar Şifreleme Algoritması



Sezar Şifreleme Algoritması

```

C:\BC5\BIN\ornekler\hafta7\or1.cpp
/* Sezar sifreleme algoritmasi */
#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char mesaj[50];
    int anahtar,i;
    printf("...: Sezar Sifreleme Algoritmasi :..\n");
    printf("\nMesaj: ");scanf("%s",&mesaj);strupr(mesaj);
    printf("\nAnahtar: ");scanf("%u",&anahtar);
    for (i=0;i<strlen(mesaj);i++) mesaj[i]=(char) (((int)mesaj[i]-65+anahtar)%26+65);
    printf("\nSifrelenmis mesaj: %s",mesaj);
    getch();
}

```

```

C:\BC5\BIN\ornekler\hafta7\or1.exe
...: Sezar Şifreleme Algoritmasi :..
Mesaj: Bursa
Anahtar: 2
Şifrelenmis mesaj: DWTUC

```

Palindrom(palindromik) kelime

Baştan ve sondan okunuşları aynı olan kelimeler, "*palindrom kelime*" olarak adlandırılmaktadır.

```

C:\BC5\BIN\ornekler\hafta7\or2.cpp
/* Palindrom kelime */
#include <stdio.h>
#include <string.h>
#include <conio.h>
main()
{
    char a[30],b[30];
    clrscr();printf(">>> Palindrom kelimeler <<<\n");
    printf("\nKelime: ");scanf("%s",&a);
    strcpy(b,a);strrev(a);
    if (strcmp(a,b)==0) printf("\nPalindrom kelimedir");
    else printf("\nPalindrom kelime degildir");
    getch();
}

```

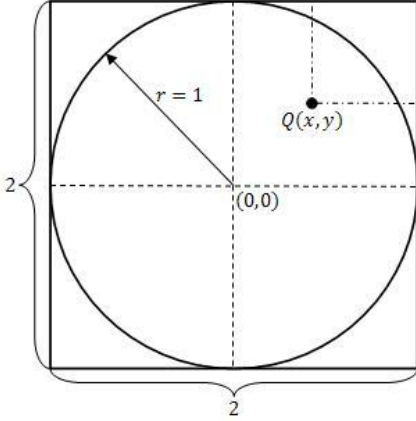
```

C:\BC5\BIN\ornekler\hafta7\or2.exe
>>> Palindrom kelimeler <<<
Kelime: Elektronik
Palindrom kelime degildir

```

Monte Carlo yöntemiyle π sayısı

Rastgele sayılardan faydalanılarak gerçekleştirilen istatistiksel modelleme ve benzetimler, genel olarak “Monte Carlo” yöntemleri olarak adlandırılmaktadır.



Bir kenarının uzunluğu 2 birim olan bir karenin içine 1 birim uzunluğunda yarıçapa sahip ve karenin kenarlarına teğet daire çizilirse; rastgele bir $Q(x, y)$ noktasının dairenin içine düşme (içinde olma) olasılığı

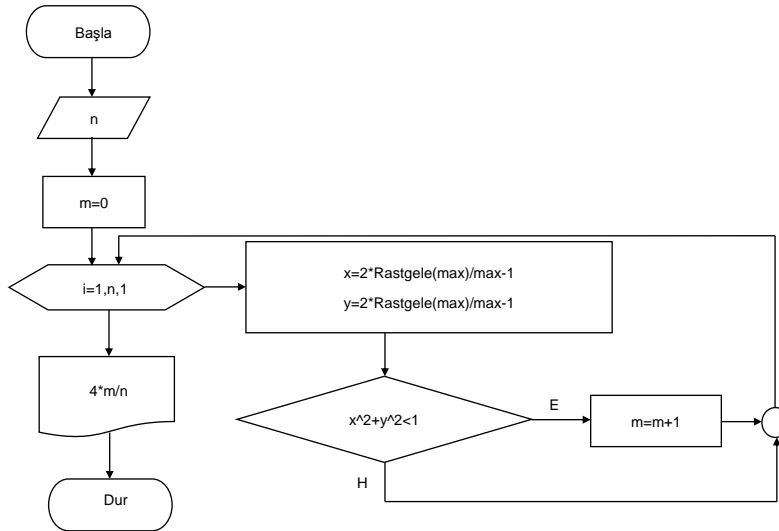
$$P(x^2 + y^2 < 1) = \frac{\text{Dairenin alanı}}{\text{Karenin alanı}} = \frac{\pi \cdot 1^2}{2 \cdot 2} = \frac{\pi}{4}$$

olur. Dolayısıyla n tane rastgele nokta üretilip bunlardan m tanesi dairenin içinde kalırsa

$$\frac{m}{n} = \frac{\pi}{4} \rightarrow \pi = \frac{4m}{n}$$

π sayısı elde edilir.

Monte Carlo yöntemiyle π sayısı



Monte Carlo yöntemiyle π sayısı

```

C:\ABC5\BINORNEKLER\HAFTA7\or3.CPP
/* pi sayısı - Monte Carlo */
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
main()
{
    double x,y,n,i=1,m=0;
    clrscr();printf("--- Monte Carlo yöntemiyle pi sayısı hesabi ---\n");
    printf("\nNokta sayısı (milyon): ");scanf("%lf",&n);n*=1000000;
    srand(time(NULL));
    while (i<=n)
    {
        x=(double) 2*rand()/RAND_MAX-1;y=(double) 2*rand()/RAND_MAX-1;
        if (x*x+y*y<1) m++; i++;
    }
    printf("\nMonte Carlo yönteminde %0.0lf rastgele nokta kullanılması sonucu pi= %0.20lf",n,4*m/n);
    getch();
}

```

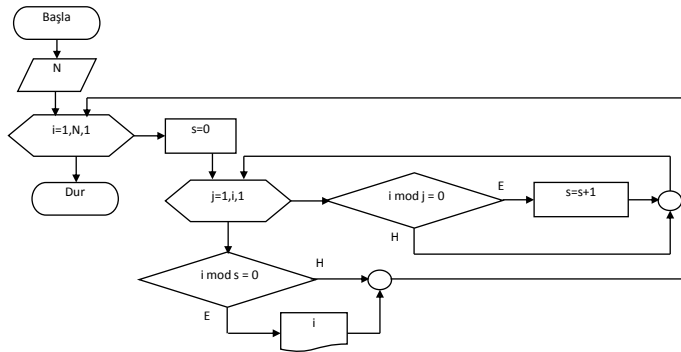
```

C:\ABC5\BINORNEKLER\HAFTA7\or3.exe
--- Monte Carlo yöntemiyle pi sayısı hesabi ---
Nokta sayısı (milyon): 1000
Monte Carlo yönteminde 1000000000 rastgele nokta kullanılması sonucu pi= 3.14137
470399999996200

```

Tau sayısı

1, 2, 8, 9, 12, 18, 24, 36, 40, 56 ... gibi tam bölenlerinin sayısına tam bölünebilen tamsayıları "*tau sayısı*" denir. Örneğin 18'in tam bölenleri 1, 2, 3, 6, 9 ve 18 olup altı adettir ve 18, 6'ya tam bölünebilmektedir.



Tau sayısı

The screenshot shows a C++ program in a Notepad window and its execution in a Command Prompt window. The program calculates and prints the first 10 Tau numbers for a given upper limit of 50.

```

C:\ABC5\BIN\ornekler\hafta7\or4.cpp
/* Tau sayisi */
#include <stdio.h>
#include <conio.h>
main()
{ int i,j,n,s;
  clrscr();printf("=== Tau sayisi ===\n");
  printf("\nÜst siniri giriniz: ");scanf("%u",&n);printf("\n");
  for (i=1;i<=n;i++)
  { s=0;for (j=1;j<=i;j++)
    if (i%j==0) s++;
    if (i%s==0) printf("%u\n",i); }
  getch();
}

C:\ABC5\BIN\ornekler\hafta7\or4.exe
=== Tau sayisi ===
Üst siniri giriniz: 50
1
2
3
4
6
8
9
12
18
24
36
40

```

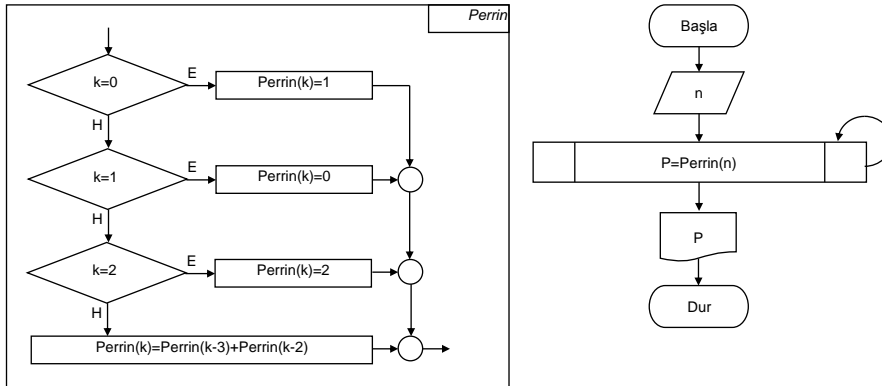
Perrin sayıları

3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17... sayıları, “*Perrin sayıları*” olarak adlandırılmaktadır.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}^k \begin{bmatrix} 3 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} P(k) \\ P(k+1) \\ P(k+2) \end{bmatrix}$$

$$P(k) = \begin{cases} 3 & , k=0 \\ 0 & , k=1 \\ 2 & , k=2 \\ P(k-3) + P(k-2) & , k > 2 \end{cases}$$

Perrin sayıları



Perrin sayıları

```

C:\BC5\BIN\ornekler\hafta7\or5.cpp
/* Perrin sayilari */
#include <stdio.h>
#include <conio.h>
int perrin(int k)
{ if (k==0) return 3;
  else if (k==1) return 0;
  else if (k==2) return 2;
  else return perrin(k-3)+perrin(k-2); }
main()
{ int i,n;
  clrscr();printf("<<< Perrin sayilari >>>\n");
  printf("\nKac tane Perrin sayisi olusturulsun? ");
  scanf("%u",&n);printf("\n");
  for(i=0;i<n;i++) printf("%u\n",perrin(i));
  getch();
}
  
```

```

C:\BC5\BIN\ornekler\hafta7\or5.exe
<<< Perrin sayilari >>>
Kac tane Perrin sayisi olusturulsun? 10
3
0
3
3
2
3
5
10
12
  
```

Grafik uygulaması



Dairenin merkezi (250,250) ve yarıçapı 100 birimdir.

```
C:\BC5\BIN\ornekler\hafta7\proj0000.c
/* Grafik uygulaması */
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
main()
{ int gd, gm;
  detectgraph(&gd, &gm); initgraph(&gd, &gm, "C:\\BC5\\BGI");
  circle(250, 250, 100);
  rectangle(150, 150, 350, 350);
  line(250, 150, 250, 350);
  line(250, 150, 150, 250);
  line(250, 150, 350, 250);
  getch(); closegraph(); return 0;
}
```

