

Lekcja 21. (pr)

Temat: Konspiracja, czyli tworzymy własne szyfry.

Cele lekcji:

Zapoznanie z cechami i zastosowaniem kluczy symetrycznych i asymetrycznych.
Wykazanie różnic w budowie i zastosowaniu kluczy publicznych i prywatnych.
Ułożenie programu wykorzystującego do szyfrowania tablice zmienne, klucze.

Uczeń:

- pokazuje na schemacie lub grafie istotę kluczy publicznych i prywatnych oraz symetrycznych i asymetrycznych
- zna i objaśnia rolę tablic w metodach szyfrowania
- układa program z kluczem generowanym losowo
- dyskutuje o strukturze ułożonego programu i przeprowadza próby jego działania

Podręcznik str. 146

Przebieg lekcji:

1. Szyfrowanie – idea i zastosowanie.
2. Rola kluczy w szyfrowaniu informacji
 - kryptologia symetryczna
 - kryptologia asymetryczna
3. **Klucz w tablicy, czyli szyfrujemy skuteczniej od Cezara**

```
#include <iostream>
using namespace std;
void szyfrowanie(char *tekst, int przesuniecie[])
{
    int skok=0;
    for (int i=0;tekst[i]!='\0';i++)
    {
        tekst[i]=(tekst[i]+przesuniecie[skok]);
        skok ++;
        if (skok ==19) skok=0;
    }
}

void odszyfrowanie(char *tekst, int przesuniecie[])
{
    int skok=0;
    for (int i=0;tekst[i]!='\0';i++)
    {
        tekst[i]=(tekst[i]-przesuniecie[skok]);
        skok ++;
        if (skok ==19) skok=0;
    }
}

int main()
{
    int klucz[]={12,13,5,3,4,6,12,19,11,10,9,8,3,7,1,14,12,17,6,2};
    cout <<endl;
```

```

char tekstdoszyfrowania[100];
cout<<" podaj tekst do szfrowania: ";
cin.get(tekstdoszyfrowania,100);
cout << "analiza tekstow: "<<endl;
cout << tekstdoszyfrowania <<endl;
szyfrowanie (tekstdoszyfrowania,klucz);
cout <<"tekst zaszyfrowany: " << tekstdoszyfrowania<<endl;
odszyfrowanie(tekstdoszyfrowania,klucz);
cout << tekstdoszyfrowania;

return 0;
}

```

4. Losowy generator, czyli dla każdego szyfrowania inny klucz

```

#include <iostream>
#include <ctime>
#include <stdlib.h>

using namespace std;

void szyfrowanie(char *tekst, int przesuniecie[])
{
    int skok=0;
    for (int i=0;tekst[i]!='\0';i++)
    {
        tekst[i]=(tekst[i]+przesuniecie[skok]);
        skok ++;
        if (skok ==19) skok=0;
    }
}

void odszyfrowanie(char *tekst, int przesuniecie[])
{
    int skok=0;
    for (int i=0;tekst[i]!='\0';i++)
    {
        tekst[i]=(tekst[i]-przesuniecie[skok]);
        skok ++;
        if (skok ==19) skok=0;
    }
}

int main()
{
    time_t tnow;
    int klucz[20],i;
    srand(time(&tnow));
    for(i=0; i<20;i++)
    {
        klucz[i]=((rand()%19)+0);
        cout<< klucz[i]<<" ";
    }
}

```

```
cout<<endl;
char tekstdoszyfrowania[100];
    cout<<"Podaj tekst do zaszyfrowania: ";
cin.get(tekstdoszyfrowania,100);
cout <<tekstdoszyfrowania<<endl;

    szyfrowanie(tekstdoszyfrowania,klucz);
cout<<tekstdoszyfrowania<<endl;
odszyfrowanie(tekstdoszyfrowania,klucz);
cout << tekstdoszyfrowania<<endl;

return 0;
}
```