

zadanie 7.1

cpp

```
int main(){
    cout<<endl<<"Witam w zadaniu 7.1"<<endl;

    fstream plik;
    plik.open("liczby.txt");
    ofstream jeden("zad_5.txt");
    string odp;
    int i;
    int n;
    float a;
    while(!plik.eof())
    {
        odp="tak";

        i=3;
        n=1;
        plik>>a;
        a=sqrt(a);
        n=a;
        if(n!=a){odp="nie";}
        if (n%2==0 && n!=2){odp="nie";}
        while(i<sqrt(n))
        {
            if(n%i==0){odp="nie";break;}
            i=i+2;
        }

        if(odp=="tak"){n=n*n;jeden<<n<<endl;}
    }
    cin.get();
    return 0;
}
```

zadanie 7.2

cpp

```
int main(){
    cout<<endl<<"Witam w zadaniu 7.2"<<endl;

    int i =0;
    string w="";
    string s="matura";
    while(i<=s.size())
    {
        w=w+s[i+1];
        w=w+s[i];
        cout<<w<<endl;
        i=i+2;
    }
    cin.get();
    return 0;
}
```

zadanie 7.3

cpp

```
int main(){
    cout<<endl<<"Witam w zadaniu 7.3"<<endl;

    int i =1;
    string s=" AYRAILTZWOGMEIWO SEYRPTN";
    string szyfr;
    int l = ceil(sqrt(s.size()));
    cout<<"l:"<<l<<endl;
    while(i<=s.size())
    {
        cout<<s[(i-l*(floor((i-1)/l))-1)*l+floor((i-1)/l)+1];
        i++;
    }
    cout<<endl<<szyfr;
    cin.get();
    return 0;
}
```

zadanie 7.4

cpp

```
bool pierwsza(int n)
{
    int i=3;
    if (n%2==0 && n!=2){return 0;}
    while(i<sqrt(n))
    {
        if(n%i==0){return 0;}
        i=i+2;
    }

    return 1;
}

int main(){
    fstream plik;
    plik.open("NAPIS.TXT");
    ofstream odp("ZADANIE5.txt");
    char a;
    char b;
    int sum=0;
    int pierwszy = 0;
    bool rise=1;
    string ros="";
    int temp;
    string word;
    vector<string> all;
    while(!plik.eof())
    {

        a=plik.get();
        word =word+a;
        sum = sum +a;
        if(a<b && a!=10){rise=0;}
        if(a==10){if(rise){ros+=word;}pierwszy+=pierwsza(sum-10);
        sum=0;rise=1;all.push_back(word);word="";}
        b=a;
    }
    cout<<pierwszy<<endl;
    cout<<ros<<endl;
    int l=all.size();int i=0;int ii=0;int iii=0;
    while(i<l)
    {
        ii=0;
        while(i+ii<l)
        {
            ii++;
            if(all[i]==all[i+ii])
            {
                odp<<all[i];
            }

        }

        i++;
    }
    cin.get();
    return 0;
}
```

zadanie 7.5

cpp

```
string szyfr(string word,string key)
{
    int l;
    int ll;
    int i;

    l=word.size();
    ll=key.size();
    i=0;
    while(i<l)
    {
        word[i]=(word[i]+key[i%ll]-128)%26+64;
        i++;
    }
    return word;
}
string deszyfr(string word,string key)
{
    int l;
    int ll;
    int i;

    l=word.size();
    ll=key.size();
    i=0;
    while(i<l)
    {
        word[i]=(word[i]-key[i%ll])+(word[i]-key[i%ll]<=0)*26+64;
        i++;
    }
    return word;
}
int main()
{
    string word ="MARTA";
    string key = "TOR";
    fstream plik;
    fstream plik2;
    plik.open("tj.txt");
    plik2.open("klucze1.txt");
    ofstream odp("wynik4a.txt");
    while(!plik.eof())
    {
        plik>>word;
        plik2>>key;
        odp<<szyfr(word,key)<<endl;
    }
    fstream plik3;
    fstream plik4;
    plik3.open("sz.txt");
    plik4.open("klucze2.txt");
    ofstream odpp("wynik4b.txt");

    while(!plik3.eof())
    {
        plik3>>word;
        plik4>>key;
        odpp<<deszyfr(word,key)<<endl;
    }
}
```

zadanie 7.6

cpp

```
int main()
{
    fstream plik;
    plik.open("slowa.txt");
    ofstream odp("wyniki6.txt");
    string temp;

    int i=0;
    int l;
    int ll;
    int jeden=0;
    int dwa =0;
    int trzy=0;
    string word;
    string t;

    while(!plik.eof())
    {
        plik>>word;
        temp=word;
        l=word.size();
        if(word[l-1]=='A'){jeden++;}
        plik>>word;
        ll=word.size();
        if(word[l-1]=='A'){jeden++;}
        if(word.find(temp,0)<100){dwa++;}
        t=word+", "+temp;
        if(word.size()==temp.size()){
            cout<<word<<", "<<temp<<endl;
            i=0;
            while(i<word.size()-1)
            {
                if(word.at(i)>word.at(i+1)){swap(word[i],word[i+1]);i=-1;}
                i++;
            }
            i=0;
            while(i<temp.size()-1)
            {
                if(temp.at(i)>temp.at(i+1)){swap(temp[i],temp[i+1]);i=-1;}
                i++;
            }
            cout<<word<<", "<<temp<<endl<<endl;
            //if(word==temp){cout<<t<<endl;}
            trzy+=(word==temp);
        }
        odp<<"6.1 "<<jeden-1<<endl<<"6.2 "<<dwa-1<<endl<<"6.3 "<<trzy-1;
        cout<<jeden-1<<endl;
        cout<<dwa-1<<endl;
        cout<<trzy-1;
    }
}
```

zadanie 8.1

cpp

```
int wartosc(string s,int p)
{
    int l=0;
    int w=1;
    int i=s.length()-1;

    int c=0;
    while(i>l)
    {
        c=s.at(i)-'0';
        l=l+w*c;
        w=w*xp;
        i--;
    }
    return l;
}
int main()
{
    int p=8;
    string s="223";
    int num=wartosc(s,p);
    cout<<endl<<num;

}
```

zadanie 8.2

cpp

```
int wartosc(string s,int p)
{
    int l=0;
    int w=1;
    int i=s.length()-1;

    int c=0;
    while(i>l)
    {
        c=s.at(i)-'0';
        l=l+w*c;
        w=w*p;
        i--;
    }
    return l;
}

int main()
{
    int jeden=0;
    int dwa=0;
    int dwadwa=0;
    int trzy=0;
    fstream plik;
    plik.open("liczby.txt");
    ofstream odp("wynik4.txt");
    string num;
    int l;
    int i;
    int temp;
    int a;
    int n;
    int min=9000000;
    int minpos=0;
    int max=0;
    int maxpos=0;
    int index=0;
    while(!plik.eof())
    {
        plik>>num;
        a=wartosc(num,2);
        if(min>a){min=a;minpos=index+1;}
        if(max<a){max=a;maxpos=index+1;}
        temp=0;
        a=0;
        l=num.size();
        i=0;
        while(i<l)
        {
            temp+=num[i]-'0';
            i++;
        }
        n=(num[l-1]-'0'+2*num[l-2]-'0')%2;
        //cout<<n<<endl;
        jeden+=(temp<l-temp);
        dwa+=(!n);
        n=(num[l-1]-'0'+2*num[l-2]-'0'+3*num[l-3]-'0')%8;
        dwadwa+=(!n);
        //cout<<a<<endl;

        index++;
    }
    cout<<jeden<<endl<<dwa<<endl<<dwadwa<<endl<<trzy<<endl;
    cout<<min<<endl<<minpos<<endl<<max<<endl<<maxpos<<endl;
    odp<<"1-"<<jeden<<endl<<"2.1-"<<dwa<<endl<<"2.2-"<<dwadwa<<endl<<"3.1-"<<minpos<<endl<<"3.2-"<<maxpos;
```