Przykładowe zadania maturalne

Liczby parzyste

W pliku **dane.txt** znajduje się w kolejnych wierszach 1000 liczb z przedziału [0;999999].

a) do pliku a.txt wpisz ilość liczb parzystych znajdujących się w pliku dane.txt w następującej postaci: "Liczb parzystych jest [ilość liczb]"

b) do pliku b.txt skopiuj wszystkie liczby z pliku dane.txt, w których cyfra dziesiątek jest równa 7 lub 0

c) do pliku c.txt skopiuj wszystkie liczby, które są kwadratami liczb całkowitych, np. taką liczbą jest liczba 225, ponieważ

```
ofstream zapis("dane.txt");
srand((unsigned)time(NULL));
int losowa;
for (int i=1;i<=1000;i++){
       losowa=rand() % 100000;
       zapis << losowa << endl;
}
zapis.close();
```

- tworzenie pliku z danymi
- ustawienie licznika pseudolosowego

- 1000 losowych liczb z zakresu 0..999999
- zapisujemy do pliku tekstowego

ZADANIE A

Ile jest liczb parzystych w pliku

```
//ZADANIE A - ile parzystych
ifstream odczyt("dane.txt");
int liczba;
int ile=0;
while (!odczyt.eof()){
  odczyt >> liczba;
  if (liczba % 2 == 0) ile++;
}
odczyt.close();
//ZAPIS
ofstream wyniki("a.txt");
wyniki << ile;
wyniki.close();
```

- plik do odczytu

- czytamy liczby, jeśli nie koniec

- jeśli liczba jest podzielna przez 2 to zwiększamy licznik

- plik a.txt do zapisu
- zapisujemy ilość liczb parzystych

ZADANIE B

Które liczby zawierają cyfrę dziesiątek równą 0 lub 7

```
//ZADANIE B - cyfra dziesiątek 7 lub 0
cout << "7 lub 0 na dziesiatkach: " << endl;
ifstream odczytb("dane.txt");
//ZAPIS
ofstream wynikib("b.txt");
string sliczba;
int len;
while (!odczytb.eof()){
  getline(odczytb, sliczba);
  len=sliczba.length();
  if (len>1)
    if (sliczba[len-2]=='7' ||
        sliczba[len-2]=='0'){
      wynikib << sliczba << endl;
      cout.width(6);
      cout << sliczba;
    }
}
odczytb.close();
wynikib.close();
```

- otwieramy plik dane.txt do odczytu
- otwieramy b.txt do zapisu
- zmienna tekstowa
- wczytujemy liczby jak napisy, bo lepiej sprawdzać drugi znak od końca w napisach

- sprawdzamy czy napis coś zawiera bo jak czyta puste na końcu do błąd
- jeśli drugi znak od końca (cyfra dziesiątek) jest równa 0 lub 7 do zapisujemy tę liczbę do b,txt

ZADANIE C

Sprawdzamy pierwiastek liczby – jeśli jest równy całkowitemu pierwiastkowi, to jest to badana liczba jest kwadratem liczby całkowitej.

```
//ZADANIE C - kwadraty liczb całkowitych
cout << "kwadraty całkowitych: " << endl;
ifstream odczytc("dane.txt");
ofstream wynikic("c.txt");
double dliczba;
while (!odczytc.eof()){
  odczytc >> dliczba;
  if ((int)sqrt(dliczba)==sqrt(dliczba)){
    wynikic << dliczba << endl;
    cout.width(10);
    cout << sqrt(dliczba) << "^2=" << dliczba;
  }
}
odczytc.close();
wynikic.close();
```

- plik dane.tx do odczytu
- plik c.txt do zapisu

- czytamy aż do końca pliku
- pobieramy liczbę
- pierwiastkujemy liczbę i sprawdzamy czy ten pierwiastek jest liczbą całkowitą
- wynik do pliku c.txt i na ekran

Przykładowe zadania maturalne

Całkowanie

Jakie jest pole powierzchni pod krzywą opisaną równaniem $y=-x^2-x+10$ w przedziale <-2, 1>.

a) Wyznacz wynik z dokładnością do 0,00001 (10000 przedziałów).
b) Oblicz pole metodą prostokątów
c) Oblicz pole powierzchni metodą trapezów
d) Porównaj oba pola powierzchni

```
double funkcja(double x) {
  return -x*x-x+10;
}

...

float xp, xk, h, calka;
xp = -2;
xk = 1;
int n=10000;

h = (xk - xp) / n;

//metoda prostokątów
calka = 0;
for (int i=1; i<=n; i++)
       calka += funkcja(xp + i*h)*h;
cout << "prostokąty: " <<  calka << endl;




//metoda trapezów
float a,b;
calka = 0;
for (int i=0; i<n; i++){
       a=funkcja(xp+i*h);
       b=funkcja(xp+(i+1)*h);
       calka += (a+b)*h/2;
}
cout << "trapezy: " <<  calka << endl;
```

- funkcja do całkowania

- zmienne pomocnicze

- przedział całkowania
- dokładność
- obliczenie szerokości przedziału

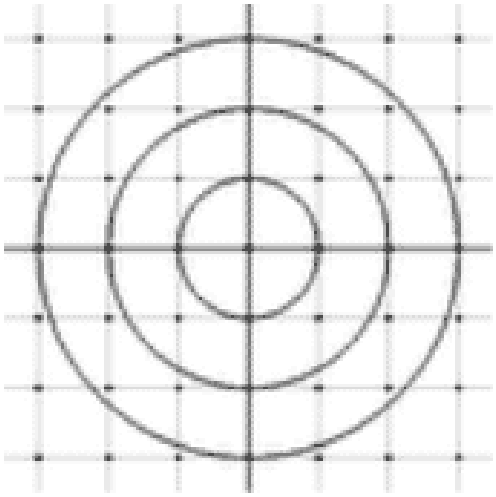
- początkowa wartość całki
- wyliczamy wartość całki dla danego X
- wartość funkcji pomnożona przez szerokość przedziału daje pole powierzchni pojedynczego prostokąta
- zwiększamy wartość pola powierzchni

- aby obliczyć pole trapezu potrzebne są dwie wysokości

- wartość funkcji na początku przedziału
- wartość funkcji na końcu przedziału
- pole trapezu

Przykładowe zadania maturalne

Punkty kratowe



Ile punktów kratowych (całkowite współrzędne) znajduje się w okręgu o promieniu R.

UWAGA. Punkt musi leżeć w okręgu, więc spełnia równanie r^2 >= x^2 + y^2.

```
float R;
cout << "Podaj promień R: ";
cin >> R;
cout << endl;
int ile=0;
float pro1,pro2;
for (int x=-R;x<=R;x++)
  for (int y=-R;y<=R;y++){
    pro1=x*x+y*y;
    pro2=R*R;
    if ( pro1 <= pro2)
      ile=ile+1;
  }
cout << "Punkty kratowe” << endl;
cout << "wewnątrz okręgu o promieniu "
     << R << " = "<< ile << endl;
```

- podaj promień z klawiatury

- x i y sterujące pętlą są całkowite,
- więc –R do +R też będzie się zmieniać całkowicie mimo tego że same są typu rzeczywistego
- wyliczamy promień okręgu z całkowitych
- wyliczamy promień okręgu podanego z klawiatury
- jeśli całkowity mniejszy to zliczamy

Przykładowe zadania maturalne

Dodawanie dużych liczb naturalnych

Napisz program, który doda do siebie dwie duże liczby naturalne.
Wejście - Dwie liczby naturalne składające się z maksymalnie 1000 cyfr.
Wyjście - Wynik suma liczb podanych na wejściu.

Dla typowych zmiennych np. integer możemy dodawać liczby całkowite które mają najwyżej kilka czy też kilkanaście cyfr. A co zrobić jeśli liczba ma tych cyfr kilkadziesiąt? Jednym ze sposobów jest potraktowanie takiej wpisanej na przykład z klawiatury liczby jako ciągu znaków. Dodając dwie takie liczby-teksty, 1) bierzemy po jednym znaku od końca, 2) zamieniamy je na cyfry, 3) dodajemy 4) zapamiętujemy przeniesienie jeśli wynik większy od 9 5) zamieniamy otrzymany wynik na znak 6) zapisujemy na odpowiedniej pozycji w tekście wyniku.

```
string liczba1,liczba2,wynik;
cout<<"Podaj pierwszą liczbę: ";
cin>>liczba1;
cout<<"Podaj drugą liczbę: ";
cin>>liczba2;
//dlugosci liczb
int dl1=liczba1.length();
int dl2=liczba2.length();
//wyznaczenie dłuższej liczby
//i uzupełnienie początku krótszej zerami
int dl;
string sroz="";
int roz=abs(dl2-dl1);
for (int i=1;i<=roz;i++){sroz=sroz+"0";}
if (dl1>dl2){
       dl=dl1;
       liczba2=sroz+liczba2;
} else{
       dl=dl2;
       liczba1=sroz+liczba1;
}
//dodawanie znak-cyfra-znak
int c1,c2,c3;
int przenies=0;
for (int i=dl-1;i>=0;i--){
       //zamiana na liczbe!!!
       c1=int(liczba1[i])-48;
       c2=int(liczba2[i])-48;
       c3=c1+c2+przenies;
       przenies=0;
       if (c3>=10){c3=c3-10;przenies=1;}
       //zamiana na tekst!!!
       wynik=char(c3+48)+wynik;
}
if (przenies==1){wynik='1'+wynik;}
//wyniki na ekran
cout.width(20);
cout << liczba1 << endl;
cout.width(20);
cout << liczba2 << endl;
cout.width(20);
cout << wynik << endl;
```

- wpisujemy z klawiatury dwie liczby
- zapamiętane w zmiennych tekstowych

- wyliczamy długości tekstów

- aby dodawać w pętli, bo wygodniej
- uzupełniamy początek krótszego tekstu zerami
- w pętli produkujemy odpowiednią liczbę zer
- i doklejamy na początku mniejszej liczby

- cały proces w pętli

- zamieniamy znak pobrany z łańcucha na liczbę

- sprawdzamy czy jest przeniesienie

- zamieniamy wynik na tekst i doklejamy na początek wyniku

- końcowe przeniesienie, bo tego pętla nie obejmowała

Przykładowe zadania maturalne

Ciągi tekstowe ABC

W pliku ciagi.txt znajduje się tysiąc wierszy, a w każdym wierszu ciąg o długości trzech liter ze zbioru {A, B, C} -litery mogą się powtarzać. Napisz program, który:

a) obliczy prawdopodobieństwo, że w wybranym wierszu będzie ciąg składającym się z takich samych znaków. Odpowiedź zapisz do pliku a.txt

b) obliczy prawdopodobieństwo, że w wybranym wierszu będzie palindrom lub litera po prawej stronie ma wartość "A". Odpowiedź zapisz do pliku b.txt.

c) skopiuje do pliku c.txt wszystkie wiersze, w których pojawił się podciąg "AC", np.:AAC, ACB, ACC, itd.

Wskazówka. W podpunkcie a i b, należy zastosować prawdopodobieństwo klasyczne ze wzoru: P(A)=|A| / |Ω|, gdzie |A| - ilość wierszy spełniających kryteria zadania, |Ω| - wszystkie wiersze.

```
//tworzenie pliku do zadania - jednorazowo
ofstream zapis("ciagi.txt");
string napis;
for (int i=1;i<=1000;i++){
       napis="";
       napis=rand() % 3+65;
       napis+=rand() % 3+65;
       napis+=rand() % 3+65;
       zapis << napis << endl;
}
zapis.close();
```

- losujemy 1000 znaków z przedziału 65..67 czyli ABC w kodzie ASCII
- i zapisujemy w pliku ciagi.txt

```
//ODCZYT
ifstream odczyt("ciagi.txt");
ofstream zapisC("c.txt");
int licznik=0;//liczymy identyczne
int licznikB=0;
int licznikC=0;
int razem=0;//liczymy łącznie
string napis;
while (!odczyt.eof()){
  odczyt >> napis;
  razem++;
  //A
  if (napis[0]==napis[1] &&
      napis[0]==napis[2] &&
      napis[2]==napis[1])
    licznik++;
  //B
  if (napis[0]==napis[2] ||
      napis[2]=='A')
  licznikB++;
  //C
  if ((napis[0]=='A' &&
       napis[1]=='C') ||
      (napis[1]=='A' &&
       napis[2]=='C')){
    zapisC << napis << endl;
    licznikC++;
  }
}
odczyt.close();
zapisC.close();
```

- takie same trzy znaki

- pierwszy znak taki sam jak ostatni lub ostatni znak równy A

- pierwszy i drugi AC albi drugi i trzeci AC

```
//RZUTOWANIE float musi być
// 4 miejsca po przecinku (3)
cout << licznik << " / " << razem
  << " A) prawdopodob=" << setprecision(3)
  << (float)licznik / razem << endl;
cout << licznikB << " / " << razem
  << " B) prawdopodob=" << setprecision(3)
  << (float)licznikB / razem << endl;
cout << licznikC << " / " << razem
  << " C) prawdopodob=" << setprecision(3)
  << (float)licznikC / razem << endl;
```

Przykładowe zadania maturalne

Pary liczb (2014)

W pliku PARY_LICZB.TXT znajduje się 1000 par liczb. Każda para jest w jednym wierszu. Liczby w parze rozdzielone są spacją. Wszystkie liczby są całkowite dodatnie, nie większe niż 30000.

Napisz program, który dla danych z pliku PARY_LICZB.TXT daje odpowiedzi do poniższych podpunktów. Odpowiedzi zapisz w pliku ZADANIE5.TXT, a każdą odpowiedź poprzedź literą oznaczającą ten podpunkt.

a) Ile jest wierszy, w których jedna z występujących tam liczb jest wielokrotnością tej drugiej?

b) Ile jest wierszy zawierających pary liczb względnie pierwszych, czyli takich, których największy wspólny dzielnik tych liczb równa się 1?

c) Ile jest wierszy, dla których suma cyfr pierwszej liczby jest równa sumie cyfr drugiej liczby?

Zadanie A

Jeśli jedna liczba jest wielokrotnością drugiej, to po pierwsze jest od niej mniejsza lub równa, a po drugie – mieści się w tej większej skończoną ilość razy (jest jej podzielnikiem). Najpierw sprawdzimy więc, która z liczb jest większa, a następnie sprawdzimy, czy dzieli się całkowicie (%)

```
ifstream odczyt("paryliczb.txt");
int liczba1,liczba2;
int licznik=0;
while (!odczyt.eof()){
  odczyt >> liczba1 >> liczba2;
  //która większa
  if (liczba2 > liczba1){
    int p=liczba2;
    liczba2=liczba1;
    liczba1=p;
  }
  if (liczba1 % liczba2 == 0)
    licznik++;
}
odczyt.close();
cout << "Zadanie A: " << licznik << endl;
```

- sprawdzamy która większa i ewentualnie zamieniamy kolejność liczb w zmiennych

- modulo liczb jest równe 0, to znaczy że się dzieli całkowicie

Zadanie B

Wykorzystamy opisaną wcześniej rekurencyjną funkcję obliczającą NWD – największy wspólny dzielnik. Jeśli NWD jest równe 1 to liczby są względnie pierwsze

```
long NWD(int a, int b){
  if (a!=b)
    if (a>b) return NWD(a-b,b);
    else return NWD(a,b-a);
  return a;
}
...
ifstream odczytb("paryliczb.txt");
int liczbab1,liczbab2;
int licznikb=0;
while (!odczytb.eof()){
  odczytb >> liczbab1 >> liczbab2;
  if (NWD(liczbab1,liczbab2)==1)
    licznikb++;
}
odczytb.close();
cout << "Zadanie B: " << licznikb << endl;
```

- rekurencyjna funkcja NWD

- plik z liczbami do czytania

- czytamy liczby do końca pliku
- jeśli NWD liczb jest równe 1
- to zwiększamy licznik

Zadanie C

Wczytane liczy zamieniamy na tablice znakowe char za pomocą funkcji itoa – tak będzie wygodniej wyodrębnić poszczególne cyfry. Każdy znak w tablicy char zamieniamy na liczbę i sumujemy.

```
ifstream odczytc("paryliczb.txt");
int liczbac1,liczbac2;
int licznikc=0;
char s1[10];
char s2[10];
while (!odczytc.eof()){
  odczytc >> liczbac1 >> liczbac2;
  itoa(liczbac1,s1,10);
  itoa(liczbac2,s2,10);
  int sum1=0;
  int sum2=0;
  for (int i=0;i<strlen(s1);i++)
    sum1=sum1+s1[i]-48;
  for (int i=0;i<strlen(s2);i++)
    sum2=sum2+s2[i]-48;
  if (sum1==sum2)
    licznikc++;
}
odczytc.close();
cout << "Zadanie C: " << licznikc << endl;
```

- tablice znakowe na dwie liczby

- czytamy dwie kolejne liczby z jednego wiersza
- itoa – zamiana liczby na łańcuch char

- sumujemy cyferki w pętlach

- sprawdzamy sumy cyfr i zwiększamy licznik gdy takie same

Przykładowe zadania maturalne

Napisy (2013)

W pliku napisy.txt znajduje się 1000 napisów o długościach od 2 do 16 znaków, każdy napis w osobnym wierszu. W każdym napisie mogą wystąpić jedynie dwa znaki: „0” lub „1”.

W wybranym przez siebie języku programowania napisz program, za pomocą którego uzyskasz odpowiedzi na poniższe polecenia. Odpowiedzi zapisz w pliku zadanie4.txt, a odpowiedź do każdego podpunktu poprzedź literą oznaczającą ten podpunkt.

a) Podaj, ile jest napisów o parzystej długości.

b) Podaj, ile jest napisów, które zawierają taką samą liczbę zer i jedynek.

c) Podaj, ile jest napisów składających się z samych zer, oraz podaj, ile jest napisów składających się z samych jedynek.

d) Dla każdej liczby k = 2, 3, ...16 podaj liczbę napisów o długości k znajdujących się w pliku napisy.txt, tzn. podaj, ile jest napisów 2-znakowych, ile jest napisów 3-znakowych itd.

```
ofstream zapis(napisy.txt");
int ile;
int znak;
for (int j=1;j<=1000;j++){
       ile=rand() %15 +2;
       string napis="";
       for (int i=1;i<=ile;i++){
              znak=rand() % 2;
              napis=napis+char(znak+48);
       }
       zapis << napis << endl;
}
zapis.close();
```

- generujemy plik z liczbami typu 01

- generujemy losowo ile cyfr w liczbie

-generujemy losowo liczbę
- zamieniamy liczbę 0 1 na znak ‘0’ ‘1’ i doklejamy

Zadanie A

```
string napis;
int suma=0;
ifstream odczyta("napisy.txt");
while (!odczyta.eof()){
       odczyta >> napis;
       if (napis.length() %2 ==0)
              suma=suma+1;
}
odczyta.close();
cout << "parzystych napisów: "
<< suma << endl << endl;
```

- jeśli długość napisu parzysta, to znaczy, że dzieli się całkowicie przez dwa

Zadanie B

```
suma=0;
int ile0,ile1;
ifstream odczytb("napisy.txt");
while (!odczytb.eof()){
       odczytb >> napis;
       int dl=napis.length();
       ile0=0;
       ile1=0;
       for (int i=0;i<dl;i++)
              if (napis[i]=='0')
                     ile0=ile0+1;
              else ile1=ile1+1;
       if (ile0==ile1)
              suma=suma+1;
}
odczytb.close();
cout << "tyle samo 0 i 1: " << suma << endl << endl;
```

- zliczamy ile jest zer i jedynek w napisie

- jeśli jest tyle samo, to zwiększamy licznik

Zadanie C

```
suma=0;
ile0=0;
ile1=0;
ifstream odczytc("napisy.txt");
while (!odczytc.eof()){
	odczytc >> napis;
	int dl=napis.length();
	suma=0;
	for (int i=0;i<dl;i++)
		suma=suma+int(napis[i])-48;
	if (suma==0) ile0=ile0+1;
	if (suma==dl) ile1=ile1+1;
}
odczytc.close();
cout << "same 0: " << ile0 << endl;
cout << "same 1: " << ile1 << endl;
```

- podsumowujemy zera i jedynki w napisie

- jeśli suma zer i jest równa zero – same zera
- jeśli suma jest równa długości – same jedynki

Zadanie D

```
int tablica[17];//od 0 do 16
for (int i=0;i<17;i++) tablica[i]=0;
ifstream odczytd("napisy.txt");
while (!odczytd.eof()){
	odczytd >> napis;
	int dl=napis.length();
	tablica[dl]=tablica[dl]+1;
}
odczytd.close();
for (int i=2;i<17;i++)
	cout << i << " " << tablica[i] << endl;
```

- tablica na ilości poszczególnych długości
- zerujemy tablicę

- sumujemy w komórkach ilości napisów o odpowiednich długościach

- wypisujemy zawartość tablicy

Przykładowe zadania maturalne

Cyfry (2012)

W kolejnych wierszach pliku cyfry.txt znajduje się 1000 liczb naturalnych, mniejszych niż 10^9 (jeden miliard), po jednej liczbie w każdym wierszu. Napisz program, który da odpowiedzi do poniższych podpunktów. Każdą odpowiedź zapisz w pliku zadanie4.txt, poprzedzając ją oznaczeniem odpowiedniego podpunktu.

a) Ile liczb parzystych jest w pliku cyfry.txt?

b) Podaj liczbę z pliku cyfry.txt, której suma cyfr jest największa oraz liczbę z tego pliku, której suma cyfr jest najmniejsza. W obu przypadkach jest tylko jedna taka liczba.

Przykład:
Dla danego zbioru liczb:
121324
66562
675100
1187010
odpowiedzią są liczby: 66562 oraz 121324, ponieważ suma cyfr liczby 66562 jest równa 25 (6+6+5+6+2) i jest największą taką sumą, zaś suma cyfr liczby 121324 (1+2+1+3+2+4) jest równa 13 i jest najmniejszą taką sumą.

c) Wypisz wszystkie liczby z pliku cyfry.txt, których cyfry tworzą ciąg rosnący.

Przykład:
Cyfry liczby 123579 tworzą ciąg rosnący, ponieważ 1<2<3<5<7<9.
Cyfry liczby 1232 nie tworzą ciągu rosnącego, ponieważ ostatnia cyfra (2) nie jest większa od przedostatniej (3).
Cyfry liczby 34556 nie tworzą ciągu rosnącego, ponieważ cyfra trzecia (5) i cyfra czwarta (5) są sobie równe.
(plik w pary liczb)

Zadanie A

```
//a) Ile liczb parzystych jest w pliku cyfry.txt?
long int liczbaa;
int ilea=0;
ifstream odczyta("cyfry.txt");
while (!odczyta.eof()){
       odczyta >> liczbaa;
       if (liczbaa % 2 == 0) ilea++;
}
odczyta.close();
cout << "liczba parzystych: " << ilea << endl;
```

- zwyczajne modulo 2 do sprawdzania parzystości

Zadanie B

```
//b) Podaj liczbę z pliku cyfry.txt, której suma cyfr jest największa
//oraz liczbę z tego pliku, której suma cyfr jest najmniejsza.
//W obu przypadkach jest tylko jedna taka liczba.
int SumaCyfr(int liczba){
    int l=liczba;
    int suma=0;
    do {
       suma = suma + l % 10;      //sumujemy reszty z dzielenia
       l = l / 10;                //kolejny wyraz zmniejszony o 10
    } while (l != 0);
    return suma;
}
...
long int liczbab;
long int Lminb=999999999;  //liczby
long int Lmaxb=0;
int Sminb=999999999;       //sumy
int Smaxb=0;

ifstream odczytb("cyfry.txt");
while (!odczytb.eof()){
       odczytb >> liczbab;
       //korzystamy z funkcji SumaCyfr
       int sumab=SumaCyfr(liczbab);
       if (sumab<Sminb) {Lminb=liczbab; Sminb=sumab;};
       if (sumab>Smaxb) {Lmaxb=liczbab; Smaxb=sumab;};
}
odczytb.close();
cout << "najmniejsza suma cyfr: " << Sminb << " " << Lminb << endl;
cout << "największa suma cyfr:  " << Smaxb << " " << Lmaxb << endl;
```

Zadanie C

```
//od tyłu maleją liczby
//od przodu rosną
bool malejacy(int liczba){
    int l=liczba;
    int r;
    bool wynik=true;
    do {
        r=l%10;
        l=l/10;
             if (l%10 >= r) wynik=false;
    } while (l!=0);
    return wynik;
}
...
//c) Wypisz wszystkie liczby z pliku cyfry.txt,
//których cyfry tworzą ciąg rosnący.
cout << "ciągi rosnące:" << endl;
long int liczbac;
ifstream odczytc("cyfry.txt");
while (!odczytc.eof()){
       odczytc >> liczbac;
       if (malejacy(liczbac)) cout << liczbac << endl;
}
odczytc.close();
```

-malejące od tyłu to to samo co rosnący od przodu

- obliczamy resztę
- obliczamy kolejny mniejszy 10 razy wyraz
- jeśli reszta z kolejnego wyrazu jest >= to nie jest malejący (od tyłu)

Przykładowe zadania maturalne

Hasła (2011)

Informatyk z firmy „KompOK” zapisał w pliku hasla.txt 200 haseł. Każde hasło umieszczone jest w osobnym wierszu pliku. Hasło składa się tylko z małych liter alfabetu angielskiego, zaś jego długość wynosi od 3 do 10 znaków.

Wykorzystując dane zawarte w tym pliku, wykonaj poniższe polecenia. Odpowiedzi do poszczególnych podpunktów zapisz w plikach tekstowych o nazwach wynik4a.txt, wynik4b.txt, wynik4c.txt.

a) W pliku wynik4a.txt podaj, ile haseł ma parzystą, a ile nieparzystą liczbę znaków.

b) W pliku wynik4b.txt utwórz zestawienie haseł (po jednym w wierszu), które są palindromami. Palindrom to wyraz brzmiący tak samo przy czytaniu z lewej strony do prawej, jak i odwrotnie, np. kajak, potop.

c) Zapisz w pliku wynik4c.txt zestawienie haseł (po jednym w wierszu) zawierających w sobie dwa kolejne znaki, których suma kodów ASCII wynosi 220.

Przykłady:
Hasło krzysio zawiera dwa kolejne znaki si, których suma kodów ASCII wynosi 220. Kod ASCII znaku s to 115, kod znaku i to 105; suma kodów wynosi 115+105 = 220.
Hasło cyrk zawiera również takie dwa kolejne znaki. Kod ASCII znaku c to 99, kod ASCII znaku y to 121; suma kodów wynosi 99+121=220
Tabela kodów ASCII
a b c d e f g h i j k l m n o p q r s t u v w x y z
97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122
Uwaga: Kolejność haseł w plikach wynik4b.txt, wynik4c.txt powinna być zgodna z kolejnością ich występowania w pliku hasla.txt.

```
//generator pliku z hasłami
//3-10 małych liter (ASCII 97-122)
ofstream zapis("hasla.txt");
string napis;
for (int i=1;i<=200;i++){
	napis="";
	//ile od 3 do 10
	int ile=rand() % 8 +3;
	for  (int j=1;j<=ile;j++){
		napis+=rand() % (122-97+1)+97;
	}
	zapis << napis << endl;
}
zapis.close();
```

- generujemy hasła z małych liter alfabetu

Zadanie A

```
int PA=0;//parzyste
int NP=0;//nieparzyste
ifstream odczytA("hasla2011.txt");
string napisA;
while (!odczytA.eof()){
	odczytA >> napisA;
	if (napisA.length() % 2 ==0)
		PA++; else NP++;
}
odczytA.close();
cout << "   parzyste: " << PA << endl;
cout << "nieparzyste: " << NP << endl;
```

- zliczamy parzyste i nieparzyste długości znaków w napisie

Zadanie B

```
cout << "PALINDROMY" << endl;
ifstream odczytB("hasla2011.txt");
string napisB;
while (!odczytB.eof()){
	odczytB >> napisB;
	int dl=napisB.length();
	int dl2=dl/2;
	bool palindrom=true;
	for (int i=0;i<dl2;i++)
		if (napisB[i]!=napisB[dl-1-i])
			palindrom=false;
	if (palindrom)
		cout << napisB << endl;
}
odczytB.close();
```

- wczytujemy kolejny napis
- wyliczamy długość
- oraz połowę długości
- pętla tylko do połowy napisu
- bo drugą połowę bierzemy od tyłu
- i sprawdzamy kolejne literki
- jeśli się nie zgadzają to nie jest palindromem

Zadanie C

```
cout << "ASCII 220"<< endl;
ifstream odczytC("hasla2011.txt");
string napisC;
while (!odczytC.eof()){
	odczytC >> napisC;
	int dl=napisC.length();
	for (int i=0;i<dl-2;i++)
		if (napisC[i]+napisC[i+1]==220)
			cout << napisC << endl;
}
odczytC.close();
```

- sprawdzamy dwie kolejne litery i sumujemy KODY
- dl-2 bo ostatniej litery nie ma z czym sprawdzać
- jeśli suma kodów daje 220 to zliczamy

Przykładowe zadania maturalne

Palindromy (2010)

Palindromem nazywamy słowo, które czytane od lewej i od prawej strony jest takie samo. Na przykład palindromami są słowa: JABFDFBAJ, HAJAHAJAH, ABBA, Słowo JANA nie jest palindromem.

W pliku dane.txt umieszczono w kolejnych wierszach 1000 słów o długościach od 2 do 25 znaków, składających się z wielkich liter A, B, C, D, E, F, G, H, I, J. Napisz program, który przegląda słowa zapisane w pliku dane.txt i wypisuje te z nich, które są palindromami, po jednym w wierszu. Kolejność wypisywania palindromów powinna być taka sama jak w pliku z danymi. Wyniki zapisz w pliku zadanie4.txt.

```
cout << "PALINDROMY" << endl;
ifstream odczytB("palindromy2010.txt");
string napisB;
while (!odczytB.eof()){
       odczytB >> napisB;
       int dl=napisB.length();
       int dl2=dl/2;
       bool palindrom=true;
       for (int i=0;i<dl2;i++)
              if (napisB[i]!=napisB[dl-1-i])
                     palindrom=false;
       if (palindrom)
              cout << napisB << endl;
}
odczytB.close();
```

- wyznaczamy połowę długości i sprawdzamy pierwszą literę w napisie z pierwszą od końca, itd., drugą i drugą od końca

- jeśli się różnią, to napis nie jest palindromem