

Operacje wyjścia

Formatowanie tekstu na ekranie

Zadanie 1. Napisz program, który wypisze poniższy tekst.
Do obliczenia sześciątów użyj operacji mnożenia.

Oto szesciany wybranych liczb:

k	k*k*k
21	9261
32	32768
-53	-148877

Koniec programu.

Zadanie 2. Napisz program, który wypisze poniższy tekst.
Do obliczenia sześciątów użyj operacji mnożenia.

Oto szesciany wybranych liczb:

k	k*k*k
21	9261
32	32768
-53	-148877

Koniec programu.

Zadanie 3. Napisz program, który wypisze poniższy tekst.
Do obliczenia sześciątów użyj operacji mnożenia.

Oto szesciany wybranych liczb
zapisane w systemie osemkowym:

k	k*k*k (system osemkowy)
2	10
3	33
5	175

Koniec programu.

Zadanie 4. Napisz program, który wypisze poniższy tekst.
Do obliczenia pierwiastków użyj funkcji `sqrt`. Pamiętaj o dołączeniu pliku `math.h`.

Oto pierwiastki kwadratowe wybranych liczb:

k	sqrt(k)
21	4.582576
32	5.656854
53	7.280110

Koniec programu.

Zadanie 5. Napisz program wypisujący poniższy tekst.

Oto pierwiastki kwadratowe wybranych liczb:

k	sqrt(k)
21	4.582575694955839800
32	5.656854249492380600
53	7.280109889280518100

Koniec programu.

Zadanie 6. Napisz program wypisujący poniższy tekst.

Oto pierwiastki kwadratowe wybranych liczb:

k	sqrt(k)
21	4.582576e+000
132	1.148913e+001
11153	1.056078e+002

Koniec programu.

Zadanie 7. Napisz program, który wypisze na ekran pierwiastki kwadratowe liczb 0.000000001, 0.00000001, 9876543210123, 987654321012. Zastosuj znak przekształcenia g.

Zadanie 8. Napisz program, który wypisze na ekran liczby całkowite -11, 0, 11111 umieszczone w jednej kolumnie. Zastosuj najpierw modyfikator +, a potem -.

Zadanie 9. Napisz program, który wypisze na ekran liczby całkowite -11, 0, 11111 umieszczone w jednej kolumnie i zajmujące pole rozmiaru 10. Zastosuj modyfikator 0.

Zadanie 10. Stosując funkcję `putchar` wypisz na ekran poniższy tekst

```
x:  f(x):
3   5
```

Niebezpieczeństwa związane z funkcją `printf`, stałe liczbowe

Zadanie 11. Napisz program, który wypisze na ekranie liczbę -5 na sześć sposobów poprzez zastosowanie znaków przekształcenia d, o, u, f, e, g. Spróbuj też użyć znaku przekształcenia s.

Zadanie 12. Napisz program, który wypisze na ekranie liczbę -5.0 na sześć sposobów poprzez zastosowanie znaków przekształcenia d, o, u, f, e, g. Spróbuj też użyć znaku przekształcenia s.

Zadanie 13. Napisz program, który wypisze na ekranie wyniki działań arytmetycznych: $2 + 3$, $2 - 3$, $2 * 3$, $3/2$, $2/3$ na następujące sposoby:

- oba argumenty w działaniach są zmiennopozycyjne (liczby w postaci 2.0, 3.0), format wypisywania wyniku zmiennopozycyjny (znak przekształcenia f lub g),

2. oba argumenty w działaniach są całkowite (liczby w postaci 2, 3), format wypisywania wyniku całkowity (znak przekształcenia d lub i),
3. jeden z argumentów w działaniach jest całkowity, drugi zmiennopozycyjny, format wypisywania wyniku zmiennopozycyjny,
4. jeden z argumentów w działaniach jest całkowity, drugi zmiennopozycyjny, format wypisywania wyniku całkowity.

Dla czytelności programu dobrze jest oprócz wyniku wypisać użytkownikowi argumenty oraz odpowiedni znak symbolizujący wykonywane działanie arytmetyczne. W tym miejscu należy dla argumentów dobrać taki znak formatowania, jakiego są one typu (dla 2 użyć %d, a dla 2.0 użyć np. %.1f).

Zadanie 14. Które z następujących instrukcji są poprawne? Przetestuj, co się stanie, gdy zamieścisz je (każdą osobno) w programie.

```
printf("%s\n", "ala");
printf("%s\n", 'ala');
printf("%s\n", ala);
printf("%s\n" "ala");
printf("%s\n", "5");
printf("%s\n", '5');
printf("%s\n", 5);
```

Sekwencje specjalne znaków

Zadanie 15. W poniższym programie

```
#include<stdio.h>
```

```
main(){
    printf("Sekwencja specjalna\ntekst za nia.");
    return 0;
}
```

przetestuj, co się stanie, gdy zastąpimy sekwencję specjalną \n następującymi sekwencjami:

```
\t, \v, \b, \r, \f, \a, \\\, \?, \', \".
```

Zadanie 16. Napisz program, który wypisze na ekranie następujący tekst, po którym nastąpi krótki sygnał dźwiękowy.

Używanie sekwencji specjalnych.

```
////////////////////////////////////
Czy "a" oznacza w C to samo co 'a'?
////////////////////////////////////
```

Koniec.

Zmienna i jej typ, operacje wejścia, instrukcje if i if-else, biblioteka math.h

Pojęcie zmiennej, pobieranie wartości od użytkownika

Zadanie 17. Napisz program, który prosi użytkownika o podanie liczby całkowitej i wypisuje ją na ekranie. Sprawdź,

co się stanie, jeśli podane dane nie będą prawidłowe (np. asf, 9.45, a45b itp.), albo liczba będzie bardzo duża (np. 123456789123456789).

Zadanie 18. Napisz program, który prosi użytkownika o podanie liczby wymiernej w postaci rozwinięcia dziesiętnego i wypisuje ją na ekranie. Sprawdź działanie programu dla danych: -5, 9.45, 234.234, 4.56e4, .3, 34e, e45, -.2e, 3.3e1894, 3,2, asf, a45b, 123456789123456789123456789).

Zadanie 19. Napisz program, który prosi użytkownika o podanie dwóch liczb całkowitych i następnie wypisuje je na ekranie. Sprawdź, co się stanie, jeśli podane dane nie będą prawidłowe (np. zamiast pierwszej liczby zostanie podane asf, 4a, 9.45, a45b itp.).

Instrukcje if, if-else i return

Zadanie 20. Napisz program, który prosi użytkownika o podanie dwóch liczb całkowitych i następnie wypisuje je na ekranie. Jeśli dane nie są prawidłowe program powinien poinformować o tym użytkownika i zakończyć działanie. Zastosuj instrukcję if. Czy w ten sposób program rozpozna wszystkie sytuacje błędne?

Zadanie 21. Napisz program, który prosi użytkownika o podanie liczby całkowitej w systemie szesnastkowym i wypisuje ją na ekranie w systemie dziesiętnym. Jeśli dane nie są prawidłowe, program powinien poinformować o tym użytkownika i zakończyć działanie. Czy liczby ujemne są wypisywane prawidłowo?

Zadanie 22. Napisz program, który prosi użytkownika o podanie liczby całkowitej w systemie dziesiętnym i wypisuje ją na ekranie w systemie szesnastkowym. Jeśli dane nie są prawidłowe, program powinien poinformować o tym użytkownika i zakończyć działanie. Czy liczby ujemne są wypisywane prawidłowo?

Zadanie 23. Napisz program, który prosi użytkownika o podanie liczby całkowitej nieujemnej w systemie dziesiętnym i wypisuje ją na ekranie w systemie szesnastkowym. Pamiętaj o kontroli poprawności danych.

Zadanie 24. Napisz program, który prosi użytkownika o podanie liczby całkowitej w systemie dziesiętnym i

- jeśli jest dodatnia, to pisze, że tak jest,
- w przeciwnym przypadku informuje, że jest to liczba niedodatnia.

Pamiętaj o kontroli poprawności danych.

Zadanie 25. Napisz program, który prosi użytkownika o naciśnięcie klawisza. Następnie informuje, czy wciśnięto t, czy inny klawisz. Rozpatrz przypadek wpisania np. tre. Czy jest to sytuacja błędna? Czy sytuacją błędną jest naciśnięcie T? Czy wpisanie spacji za znakiem jest dopuszczalne? Napisz różne wersje tego programu, w zależności od przyjętych ustaleń, co jest błędem.

Zadanie 26. Napisz program, który rozstrzyga, czy punkt o współrzędnych podanych przez użytkownika należy do koła o środku (0,0) i promieniu 1. Pamiętaj o kontroli poprawności danych.

Zadanie 27. Napisz program, który rozstrzyga, czy punkt o współrzędnych podanych przez użytkownika leży

1. na okręgu o środku (0,0) i promieniu 1,
2. wewnątrz tego okręgu,
3. na zewnątrz tego okręgu.

Pamiętaj o kontroli poprawności danych.

Zadanie 28. Napisz program, który rozstrzyga, czy punkt o współrzędnych podanych przez użytkownika leży w pierwszej ćwiartce układu współrzędnych. Pamiętaj o kontroli poprawności danych.

Zadanie 29. Napisz program, który informuje, w której ćwiartce, albo na której osi, leży punkt o współrzędnych podanych przez użytkownika. Pamiętaj o kontroli poprawności danych.

Funkcje z biblioteki math.h

Zadanie 30. Napisz program, który wczytuje liczbę wymierną w postaci rozwinięcia dziesiętnego i

- jeśli jest całkowita, to pisze, że tak jest,
- w przeciwnym przypadku zaokrągla ją w dół do najbliższej liczby całkowitej i wypisuje wynik tego zaokrąglenia.

Pamiętaj o kontroli poprawności danych.

Zadanie 31. Dana jest funkcja

$$f(x) = \sqrt{\left| \arccos(x) + \frac{e^x}{\ln(x)} \right|}.$$

Napisz program, który rozstrzyga, czy punkt o współrzędnych podanych przez użytkownika

1. ma odciętą nie należącą do dziedziny funkcji $f(x)$,
2. jest punktem wykresu funkcji $f(x)$,
3. leży nad wykresem funkcji $f(x)$,
4. leży pod wykresem funkcji $f(x)$.

Wartość funkcji powinna być obliczana tylko jeden raz. Pamiętaj o kontroli poprawności danych.

Modyfikacja wartości zmiennych, instrukcje warunkowe (cd.), biblioteka ctype.h

Operatory przypisania, operatory zwiększania i zmniejszania

Zadanie 32. Przeanalizuj poniższy program i zapisz na kartce, jakie liczby spodziewasz się zobaczyć na ekranie. Następnie wykonaj program i sprawdź swoje wyniki.

```
#include<stdio.h>

int main(){
    int x;

    printf("Testowanie instr. przypisania.\n\n");
    x=3;
    printf("Po x=3;      wart. x jest: %d\n",x);
    x=5;
    printf("Po x=5;      wart. x jest: %d\n",x);
    x=x+2;
    printf("Po x=x+2;     wart. x jest: %d\n",x);
    x=x*(x+1);
    printf("Po x=x*(x+1); wart. x jest: %d\n",x);
    x=x+1;
    printf("Po x=x+1;     wart. x jest: %d\n",x);
    x=x-(3/4);
    printf("Po x=x-(3/4); wart. x jest: %d\n",x);
    x=x/(x+1);
    printf("Po x=x/(x+1); wart. x jest: %d\n",x);
    x--;
    printf("Po x--;      wart. x jest: %d\n",x);
    printf("\n\nKoniec programu. \n");
    return 0;
}
```

Napisz program równoważny powyższemu używając operatorów przypisania +=, -=, *=, /= w wymienionych niżej instrukcjach.

```
x=x+2;
x=x*(x+1);
x=x+1;
x=x-(3/4);
x=x/(x+1);
x--;
```

Zadanie 33. Niech x i y będą zmiennymi typu int. Które z poniższych instrukcji dają takie same wyniki?

```
y=(x++)+7; /*1*/
y=(++x)+7; /*2*/
y=7+(x++); /*3*/
y=7+(++x); /*4*/
y=7+(x+1); /*5*/
```

Napisz program testujący zachowanie tych instrukcji. Następnie zastąp instrukcje 1–4 równoważnymi, w których nie występuje operator zwiększania.

Zadanie 34. Dana jest funkcja

$$f(x) = \sqrt{x^2 + 1} + \frac{\ln(x^2 + 1)}{5 + \sin(x^2 + 1)} - e^{\sin(x^2 + 1)}.$$

Napisz program, który prosi użytkownika o podanie liczby i oblicza wartość funkcji f dla tej liczby. Postaraj się zminimalizować liczbę operacji arytmetycznych poprzez użycie zmiennych pomocniczych. Jak możesz przetestować, czy uzyskane wyniki są poprawne?

Zadanie 35. Napisz program, który dla podanej przez użytkownika liczby x oblicza x^8 . Wolno zastosować tylko trzy operacje mnożenia i jedną zmienną. Nie wolno użyć funkcji pow.

Zadanie 36. Dana jest funkcja

$$f(x) = 13(x-2) + 17(x-2)^2 + 23(x-2)^3.$$

Napisz program, który dla podanej przez użytkownika liczby x oblicza $f(x)$. Wolno zastosować tylko

- jedną operację odejmowania,
- dwie operacje dodawania,
- trzy operacje mnożenia,
- dwie zmienne.

Nie wolno użyć funkcji `pow`.

Wskazówka: przekształć wzór opisujący funkcję f .

Zadanie 37. Dana jest funkcja

$$f(x, y) = y + \sqrt{x^2 + y^2} + \frac{x^2 + y^2}{\cos(x^2)}.$$

Napisz program, który prosi użytkownika o podanie liczb x i y mniejszych co do modułu od 1 i oblicza wartość funkcji f dla nich. Postaraj się zminimalizować liczbę operacji arytmetycznych. Nie wolno stosować funkcji `pow`. Czy umiesz tak napisać program, by użyć tylko trzech zmiennych przy tej samej liczbie operacji?

Konstrukcja `else-if`, instrukcja `switch`, wyrażenie warunkowe

Zadanie 38. Napisz program, który pobiera dwie liczby całkowite i wyświetla menu umożliwiające wybór jednego z działań: dodawania, odejmowania, mnożenia i dzielenia. Po dokonaniu wyboru przez użytkownika wypisuje wynik. Pamiętaj o kontroli poprawności danych. Szczególną uwagę zwróć na operację dzielenia.

Program napisz w dwóch wersjach: raz stosując tylko instrukcję `if-else`, drugi raz używając też instrukcji `switch`.

Zadanie 39. Dana jest funkcja

$$f(x) = \begin{cases} 2^x & \text{dla } x > 0, \\ -1 & \text{w p.p.} \end{cases}$$

Napisz program, który prosi użytkownika o podanie liczby zmiennopozycyjnej x i oblicza wartość funkcji f dla tej liczby. Do obliczenia wartości funkcji f nie używaj instrukcji `if`.

Zadanie 40. Napisz program, który oblicza odległość R danego punktu (x, y) od początku układu współrzędnych. Odciętą i rzędną punktu o podaje użytkownik.

W programie należy posłużyć się dwoma algorytmami i porównać ich wyniki. Pierwszy algorytm posługuje się klasycznym wzorem

$$R = \sqrt{x^2 + y^2}.$$

Drugi korzysta z zależności równoważnej

$$R = \begin{cases} |y| & \text{gdy } x = 0, \\ |x|\sqrt{1 + (\frac{y}{x})^2} & \text{gdy } |x| \geq |y| \text{ i } x \neq 0, \\ |y|\sqrt{1 + (\frac{x}{y})^2} & \text{w p.p.} \end{cases}$$

Jeśli oba algorytmy dają ten sam wynik, na ekranie powinna pojawić się następująca informacja.

OBA ALGORYTMY DAJA TEN SAM WYNIK.

Odległość punktu (3,4) od (0,0) wynosi 5.

Jeśli wyniki są różne, użytkownikowi należy podać informację jak niżej.¹

ALGORYTMY DAJA RÓŻNE WYNIKI.

Odległość punktu (2,20) od (0,0) wynosi:

-wg "klasycznego" algorytmu 20.09975124224178,
-wg "specjalnego" algorytmu 20.09975124928470.

Wartość bezwzględna różnicy między wynikami:

7.0429244658498646e-009.

Do przechowywania danych i wyników użyj typu `double`. Zastosuj specyfikację `%lf` podczas wczytywania danych oraz `%.17g` do wypisywania wyników algorytmu.

Przetestuj działanie programu dla następujących punktów: (3, 4), (2, 20), (34, 987), (0, 10^{-154}), (0, 10^{-155}), (0, 10^{-170}), (0, 10^{-400}), (10^{154} , 0), (10^{155} , 0). Jeśli dla pewnych danych program zostaje przerywany komunikatem o błędzie „wykomentuj” instrukcje algorytmu „klasycznego” i jeszcze raz uruchom program dla tych danych. Czy umiesz wyjaśnić różnice w zachowaniu się obu algorytmów?

Zadanie 41. Napisz program, który sprawdza, czy dwa wektory swobodne o współrzędnych podanych przez użytkownika są

- prostopadłe,
- równoległe.

Osobno rozpatrz wektor zerowy.

Zadanie 42. Napisz program, który oblicza pole trójkąta. Użytkownik na początku powinien mieć możliwość wyboru, jakie dane chce podać:

1. bok i spuszczoną nań wysokość,
2. dwa boki i kąt między nimi,
3. trzy boki.

Sprawdź, czy dla podanych danych możliwe jest istnienie trójkąta. Pozostaw wybór użytkownikowi, czy kąt ma podać w stopniach czy radianach.

Zadanie 43. Napisz program, który spośród trzech liczb podanych przez użytkownika wybiera minimalną.

Stałe znakowe, kody ASCII, biblioteka `cctype.h`

Zadanie 44. Napisz program, który dla wpisanego znaku podaje jego kod ASCII.

Zadanie 45. Napisz program, który dla wpisanego kodu ASCII (należącego do zbioru podstawowego) wypisuje odpowiadający mu znak, o ile jest to znak drukowalny. W przypadku, gdy jest to znak kontrolny podaje informację o tym.

¹ Wyniki mogą zależeć od komputera. Dla punktu (20,2) niektóre komputery dla obu algorytmów dadzą ten sam wynik.

Zadanie 46. Napisz program, który prosi użytkownika o podanie godziny w formacie hh:mm (np. 03:12, 23:01). Następnie sprawdza poprawność wprowadzonych danych (musi być dokładnie pięć znaków, godziny z zakresu 0–23, zaś minuty z 0–59) oraz wypisuje informację postaci

Oto wczytana godzina:
11 minut po 23

z zachowaniem form gramatycznych. W przypadku błędnych danych program powinien precyzyjnie określić na czym polega błąd. Przed przystąpienia do tworzenia algorytmu opisz, jakie testy przeprowadzisz.

Zadanie 47. Napisz program, który prosi użytkownika o podanie daty z lat 2001–2050. Wymagany formatem jest dd.mm.rr (np. 03.12.10, 23.01.02). Następnie sprawdza poprawność wprowadzonych danych (musi być dokładnie osiem znaków, kwiecień nie może mieć 31 dni, liczba dni lutego zależy od tego, czy rok jest przestępny itp.). Na koniec wypisuje informację postaci

Oto wczytana data
11 kwietnia 2010

z zachowaniem form gramatycznych. W przypadku błędnych danych program powinien precyzyjnie określić na czym polega błąd. Przed przystąpienia do tworzenia algorytmu opisz, jakie testy przeprowadzisz.

Instrukcja while, stałe

Zadanie 48. Napisz program, który wypisuje na ekranie 50 początkowych potęg dwójki.

Zadanie 49. Napisz program, który wypisuje na ekranie wartości 1!, 2!, ..., 200!.

Zadanie 50. Napisz program, który dla $n = 1, 2, \dots, 200$ wypisuje wartości funkcji

$$f(n) = \frac{3^n}{(n+1)!}.$$

Zadanie 51. Napisz program, który znajduje najmniejszą liczbę naturalną n taką, że

$$\frac{90^n}{(n+1)!} < 1$$

Zadanie 52. Napisz program, który prosi o podanie jednej linii tekstu, a następnie przepisuje ją poniżej, przy czym usuwa zbędne spacje (tzn. z grupy stojących obok siebie dwóch lub więcej spacji pozostawia jedną).

Zadanie 53.² Napisz program, który prosi użytkownika o podanie liczby naturalnej. Robi to tak długo, aż dane będą poprawne (tzn. wpisana zostanie liczba naturalna, a po niej ENTER). Potem wypisuje wczytaną liczbę na ekran.

Zadanie 54.³ Napisz program, który prosi użytkownika o naciśnięcie klawisza **t**. Robi to tak długo, aż dane będą poprawne (tzn. naciśnięte zostanie **t**, a po nim ENTER). Potem wypisuje stosowny komunikat.

²Porównaj program 17.

³Porównaj program 25.

Zadanie 55. Napisz program, który wczytuje ciąg liczb całkowitych dodatnich i wypisuje minimalną. Decyzję o liczbie elementów ciągu podejmuje użytkownik: gdy chce zakończyć podawanie danych wpisuje 0. Użytkownik powinien móc poprawić każdą błędnie wprowadzoną daną (bez zaczynania od początku).

Zadanie 56. Napisz program, który wczytuje 10 liczb zmien-nopozycyjnych i wypisuje ich średnią arytmetyczną. Użytkownik powinien móc poprawić każdą błędnie wprowadzoną daną (bez zaczynania od początku).

Zadanie 57. Napisz program, który wczytuje ciąg liczb całkowitych dodatnich i wypisuje ich średnią arytmetyczną i geometryczną. Decyzję o liczbie elementów ciągu podejmuje użytkownik: gdy chce zakończyć podawanie danych wpisuje 0. Użytkownik powinien móc poprawić każdą błędnie wprowadzoną daną (bez zaczynania od początku).

Zadanie 58. Napisz program, który wczytuje ciąg zer i jedynek. Długość ciągu nie przekracza 32 znaków, ale może być dowolna. Ciąg ten jest interpretowany jako reprezentacja liczby naturalnej w systemie dwójkowym. Program ma wypisać tę liczbę w systemie dziesiętnym.

Zadanie 59.⁴ Napisz program, który wypisuje menu

0. Zakoncz program
1. Wczytaj liczby
2. Dodaj
3. Odejmij
4. Pomnoz
5. Podziel.

Po wybraniu i wykonaniu stosownej akcji program znów wypisuje menu i można ponownie dokonać wyboru (o ile poprzednim życzeniem nie było zakończenie programu). Operacje arytmetyczne są wykonywane dla zadanych przez użytkownika liczb. Zadbaj, by można było poprawić błędnie wprowadzone dane. Rozpatrz sytuację, gdy użytkownik nie podał liczb, a chce wykonać np. dodawanie.

Zadanie 60. Zmień program 42 następująco:

1. wzbogać menu o opcję zakończenia programu,
2. po dokonaniu przez użytkownika wyboru i zakończeniu wszystkich akcji związanych z tym wyborem znów powinno ukazać się menu (za wyjątkiem sytuacji wybrania zakończenia programu),
3. użytkownik może poprawić błędnie wprowadzone dane.

Zadanie 61. Zmień program 46 tak, by użytkownik mógł poprawić błędnie wprowadzone dane.

Zadanie 62. Napisz program, który prosi użytkownika o podanie swego imienia. Następnie sprawdza, czy

1. imię nie zawiera innych znaków niż litery,
2. imię zaczyna się wielką literą,
3. poza pierwszą literą wszystkie pozostałe są małe.

⁴Porównaj Program 38.

Jeśli dane są nieprawidłowe, to użytkownik zostaje o tym powiadomiony i ponownie poproszony o podanie imienia. Oto przykładowy dialog.

```
Jak masz na imie?
agata
Imie należy pisac wielka litera.

Jak masz na imie?
Ag12ata
Masz na imie Ag...?
Imie sklada sie tylko z liter.

Jak masz na imie?
AGAta
Masz na imie A...?
Poza pierwsza litera wszystkie powinny byc male.

Jak masz na imie?
Agata
Masz na imie Agata.

Koniec programu.
```

- Od tego momentu wybrane programy należy pisać z podziałem na funkcje (ustala to prowadzący zajęcia).
- Początkowo może być konieczność pisanie części programów w dwóch wersjach: z podziałem na funkcje i bez.
- Wszystkie programy używające tablic muszą być pisane z podziałem na funkcje.

Instrukcja for

Zadanie 63. Napisz program równoważny poniższemu zastępując pętlę `while` instrukcją `for`.

```
#include<stdio.h>
#include<math.h>

#define POCZ 0.0          //poczatek przedzialu
#define KON  0.125       //koniec przedzialu
#define ILE  33           //liczba punktow

int main(){
    double krok=(KON-POCZ)/(ILE-1); //odl. miedzy pkt.
    int i=0;

    printf("Program oblicza bledy przyblizenia");
    printf(" funkcji sin(x) przez x\ndla %d", ILE);
    printf(" rownoodleglych punktow z przedzial");
    printf("u [%.g,%.g] .\n\n%-14s", POCZ, KON, "x");
    printf("%-13s%-12s", "sin(x)", "x-sin(x)");
    printf("%-12s\n", "|(x-sin(x))/x|");

    while (i<=ILE){
        double x=POCZ+i*krok;
        double sin_x=sin(x);
        double bl_b=x-sin_x;
        double bl_w=x!=0.0?fabs(bl_b/x):-1.0;
        printf("\n%11.9f%14.9f%12.3e", x, sin_x, bl_b);
        if (bl_w!=-1.0)
            printf("%12.3e", bl_w);
        i++;
    }
    printf("\n\nKoniec programu.\n");
}
```

Zadanie 64. W pliku nagłówkowym `limits.h` zdefiniowana jest stała `INT_MAX` oznaczająca największą liczbę typu `int`. Napisz program, który wypisze 10 kolejnych liczb naturalnych zaczynając od `INT_MAX - 5`.

Zadanie 65.⁵ Napisz program, który wypisuje n początkowych potęg dwójki. Użyj pętli `for`. Liczbę naturalną n podaje użytkownik.

Zadanie 66. Napisz program 56 używając tam, gdzie jest to uzasadnione, pętli `for`.

Zadanie 67. Napisz program, który rysuje na ekranie tyle gwiazdek w jednej linii, ile zażyczy sobie użytkownik.

Zadanie 68. Napisz program, który rysuje na ekranie gwiazdki ułożone w „kwadrat”.

```
***
***
***
```

Liczbę gwiazdek w jednym rzędzie podaje użytkownik.

Zadanie 69. Napisz program, który rysuje na ekranie gwiazdki ułożone w prostokąt o szerokości dwa razy mniejszej od długości.

⁵Porównaj program 48.

```
*****
*****
*****
```

Liczbę rzędów podaje użytkownik.

Zadanie 70. Napisz program, który rysuje na ekranie gwiazdki ułożone w trójkąt poniższego typu.

```
*
**
***
****
*****
```

Liczbę rzędów podaje użytkownik.

Zadanie 71. Napisz program, który rysuje na ekranie gwiazdki ułożone w trójkąt poniższego typu.

```
  *
 ***
*****
*****
```

Liczbę rzędów podaje użytkownik.

Zadanie 72. Napisz program, który wypisuje tabliczkę mnożenia w zakresie do piętnastu. Sposób wypisywania ma być analogiczny do poniższego (tu zakres wynosi 5).

```
      1   2   3   4   5
-----
1 |  1   2   3   4   5
2 |  2   4   6   8  10
3 |  3   6   9  12  15
4 |  4   8  12  16  20
5 |  5  10  15  20  25
```

Zadanie 73. Napisz program, który wypisze tabelę znaków odpowiadających kolejnym kodom ASCII. Uwzględnij kody od 0 do 255. Poniższe znaki kontrolne powinny być przedstawione symbolicznie.

	kod	skrót	kod	skrót
	0	nul	12	np
	7	bel	13	cr
	8	bs	26	eof
	9	ht	27	esc
	10	nl	127	del
	11	vt		

Zadanie 74. Napisz program, który wypisze sumę w postaci poniższego tekstu

$$1 + 22 + 333 + 4444 + \dots + \underbrace{nn \dots n}_n$$

oraz jej wynik. Liczbę $n \in \{1, 2, \dots, 9\}$ podaje użytkownik.

Instrukcja for — cd., tablice jednowymiarowe

Zadanie 75. Napisz program, który pobiera od użytkownika 10 liczb całkowitych i wypisuje je w odwrotnej kolejności (tzn. od ostatnio wczytanej do pierwszej).

Zadanie 76. Napisz program, który wyznacza iloczyn oraz sumę niezerowych elementów ciągu n elementowego. Ciąg ma być podany przez użytkownika. Rozwiąż zadanie na dwa sposoby: z użyciem tablicy i bez niej. Można dodatkowo założyć, że n jest nie większe niż 100.

Zadanie 77. Napisz program, który wypisuje wszystkie trzy-cyfrowe liczby, których suma sześciastów poszczególnych cyfr jest równa danej liczbie. Staraj się zminimalizować liczbę operacji arytmetycznych. Czy może pomóc w tym użycie tablicy?

Zadanie 78. Napisz program, który oblicza n wyrazów ciągu Fibbonaciego i wypisuje je od największej do najmniejszej. Liczbę naturalną n podaje użytkownik. Można dodatkowo założyć, że n jest nie większe niż 20.

Zadanie 79. Napisz program, który wczytuje 10 liczb całkowitych i wypisuje je w kolejności od najmniejszej do największej. Zastosuj algorytm, który wyszukuje najmniejszą liczbę w tablicy i przesuwa ją na początek, potem szuka drugiej co do wielkości i umieszcza na drugiej pozycji itd.

Zadanie 80. Napisz program, który losuje 10 liczb całkowitych nieujemnych i wypisuje je w kolejności od najmniejszej do największej. Zastosuj funkcje `rand` oraz `srand` z biblioteki `stdlib.h`.

Zadanie 81. Napisz program, który losuje 10 liczb z przedziału $[1, 2]$ i wypisuje je w kolejności od najmniejszej do największej. Zastosuj funkcje `rand` oraz `srand` z biblioteki `stdlib.h`.

Zadanie 82. Napisz program, który wczytuje n liczb całkowitych i sprawdza, czy wśród nich znajduje się para jednakowych. Liczbę naturalną n podaje użytkownik, można dodatkowo założyć, że jest ona nie większa niż 100.

Zadanie 83. Napisz program, który oblicza wartość wielomianu

$$w(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

w punkcie x . Stopień, współczynniki oraz x powinien podać użytkownik. Można założyć, że stopień jest nie większy niż 100. Nie wolno stosować funkcji `pow`. Czy umiesz tak napisać algorytm, by wykonywał jedynie n dodawań i n mnożeń?

Zadanie 84. Który z czterech wzorów najlepiej nadaje się do zaprojektowania algorytmu obliczającego wartość $\binom{n}{k}$?

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{\prod_{i=1}^{n-k} (k+i)}{\prod_{i=1}^{n-k} i} = \prod_{i=1}^{n-k} \frac{k+i}{i} = \prod_{i=1}^{n-k} \left(1 + \frac{k}{i}\right)$$

Zaprogramuj ten algorytm. Czy zdoła on prawidłowo obliczyć np. $\binom{29}{14} = 77558760$? Jeśli nie, to popraw algorytm. Przy pomocy tego algorytmu wypisz 35 wierszy trójkąta Pascala. Czy wszystkie są wypisane poprawnie? Ile operacji arytmetycznych potrzeba, by wypisać 25-ty wiersz? Dla celów testowania zamieszczam poniżej 13 wierszy trójkąta Pascala.

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1

```

Zadanie 85. Algorytm z poprzedniego zadania nie jest najlepszy. Skorzystaj ze znanej zależności pomiędzy dwoma sąsiednimi wierszami do opracowania nowego algorytmu. Przy jego pomocy wypisz 35 wierszy trójkąta Pascala. Ile operacji arytmetycznych potrzeba, by wypisać 25-ty wiersz?

Zadanie 86. Napisz program, który generuje poniższy wykres. Zadbaj, by łatwo móc zmienić funkcję, której wykres jest rysowany.

WYKRES FUNKCJI $f(x)=\sin(x)$ w $[0,6.28]$:

```

^
|
|      *****
|    **      **
|  **          **
|*              *
*-----*-----*-----*-->
|
|              *          *
|            **          **
|
|          **          **
|              *****
|

```

Tablice dwuwymiarowe

Zadanie 87. Napisz program wczytujący oraz wypisujący elementy tabeli, która ma m wierszy i n kolumn:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}.$$

Liczby naturalne m i n podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 88. Napisz program, który dla zadanej przez użytkownika tabeli o m wierszach i n kolumnach:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

wypisuje tabelę z zamienionymi wierszami i kolumnami:

$$\begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix}.$$

Liczby naturalne m i n podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 89. Napisz program, który znajduje element maksymalny w tabeli liczb o m wierszach i n kolumnach podanej przez użytkownika. Liczby naturalne m i n też podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 90. Napisz program, który dla tablicy o m wierszach i n kolumnach oraz elementach, które są albo zerami, albo jedynkami sprawdza, czy w każdym wierszu i w każdej kolumnie stoi dokładnie jedna jedynka. Liczby naturalne m i n oraz elementy tabeli podaje użytkownik. Można założyć, że m i n są nie większe niż 25. Program powinien nie pozwolić użytkownikowi podawać innych elementów tabeli niż 0 lub 1.

Zadanie 91. Napisz program obliczający sumę wszystkich elementów w tabeli liczb. Tabela ma m wierszy i n kolumn. Liczby naturalne m i n oraz elementy tabeli podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 92. Napisz program, który oblicza sumę elementów leżących na głównej przekątnej tabeli liczb o n wierszach i n kolumnach. Główną przekątną tabeli

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

tworzą liczby $a_{11}, a_{22}, \dots, a_{nn}$. Liczbę naturalną n oraz elementy tabeli podaje użytkownik. Można założyć, że n jest nie większe niż 25.

Zadanie 93. Napisz program, który oblicza sumę elementów leżących powyżej głównej przekątnej tabeli liczb o n wierszach i n kolumnach. Liczbę naturalną n oraz elementy tabeli podaje użytkownik. Można założyć, że n jest nie większe niż 25.

Zadanie 94. Napisz program, który sprawdza, czy wszystkie elementy tabeli o n wierszach i n kolumnach leżące poniżej głównej przekątnej są zerami. Liczbę naturalną n oraz elementy tabeli podaje użytkownik. Można założyć, że n jest nie większe niż 25.

Zadanie 95. Napisz program, który zamienia miejscami dwa wiersze tabeli o m wierszach i n kolumnach. Liczby naturalne m i n oraz elementy tabeli podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 96. Napisz program, który w tabeli o n wierszach i n kolumnach zamienia s -ty wiersz z s -tą kolumną, gdzie s jest numerem wiersza i kolumny, na przecięciu których jest umieszczony element maksymalny głównej przekątnej. Liczbę naturalną n oraz elementy tabeli podaje użytkownik. Można założyć, że n jest nie większe niż 25.

Zadanie 97. Napisz program, który dla dwóch tabel

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix},$$

o m wierszach i n kolumnach oblicza elementy trzeciej

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mn} \end{bmatrix},$$

spełniające dla $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ zależność

$$c_{ij} = a_{ij} + b_{ij}.$$

Liczby naturalne m i n oraz elementy obu tabel podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 98. Napisz program, który dla dwóch tabel

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nk} \end{bmatrix}.$$

oblicza elementy trzeciej

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mk} \end{bmatrix},$$

spełniające dla $i = 1, 2, \dots, m, j = 1, 2, \dots, k$ zależność

$$c_{ij} = \sum_{s=1}^n a_{is} b_{sj}.$$

Liczby naturalne k, m i n oraz elementy obu tabel podaje użytkownik. Można założyć, że k, m i n są nie większe niż 25.

Zadanie 99. Napisz program, który dla zadanej tabeli o m wierszach i n kolumnach:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

oblicza liczbę

$$\max_{i=1,2,\dots,m} \sum_{j=1}^n |a_{ij}|.$$

Liczby naturalne m i n oraz elementy tabeli podaje użytkownik. Można założyć, że m i n są nie większe niż 25.

Zadanie 100. Napisz program, który dla zadanej tabeli o m wierszach i n kolumnach:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

oblicza liczbę

$$\sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}.$$

Liczby naturalne m i n oraz elementy tabeli podaje użytkownik. Można założyć, że m i n są nie większe niż 25. Czy umiesz, wzorując się na programie 40, tak zaprojektować algorytm, by uniknąć nadmiaru i niedomiaru w wynikach pośrednich?

Zadanie 101. (dodatkowe) Napisz program, który dla zadanej tabeli o m wierszach i n kolumnach:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

oblicza liczby

1. $\sum_{i=1}^m \prod_{j=1}^n \sqrt{|a_{ij}|},$
2. $\sum_{j=1}^n \prod_{i=1}^m |a_{ij}|,$
3. $\prod_{i=1}^m \sum_{j=1}^n \frac{1}{2 + |a_{ij}|},$
4. $\prod_{j=1}^n \sum_{i=1}^m a_{ij},$
5. $\max_{i=1,2,\dots,m} \prod_{j=1}^n a_{ij}^3,$
6. $\max_{j=1,2,\dots,n} \prod_{i=1}^m a_{ij}^3,$
7. $\prod_{i=1}^m \max_{j=1,2,\dots,n} a_{ij}^3,$
8. $\sum_{i=1}^m \min_{j=1,2,\dots,n} \frac{1}{a_{ij}^2 + 1},$
9. $\sum_{i=1}^m \frac{1}{2 + \min_{j=1,2,\dots,n} |a_{ij}|^2},$
10. $\max_{i=1,2,\dots,m} \min_{j=1,2,\dots,n} a_{ij},$
11. $\min_{i=1,2,\dots,m} \max_{j=1,2,\dots,n} a_{ij},$
12. $\sum_{i=1}^m \prod_{j=i}^n \sqrt{|a_{ij}|},$
13. $\sum_{i=1}^m \prod_{j=1}^i \sqrt{|a_{ij}|},$ (zakładamy, że tablica jest kwadratowa),
14. $\sum_{i=2}^{m-1} \prod_{j=i-1}^{i+1} \sqrt{|a_{ij}|},$ (zakładamy, że tablica jest kwadratowa),
15. $\sum_{i=1}^m \prod_{j=1}^n \sqrt{j + |a_{ij}|},$
16. $\sum_{i=1}^m \min_{j=1,3,5,\dots,n} \frac{1}{a_{ij}^2 + i},$ (zakładamy, że n jest nieparzyste).

Liczby naturalne m i n podaje użytkownik. Można założyć, że m i n są nie większe niż 25. Elementy tabeli są liczbami całkowitymi z przedziału $[-20, 20]$ wygenerowanymi losowo.

Poniższy materiał nie obowiązuje na egzaminie. Umiejętność rozwiązywania tych zadań ułatwi Państwu zaliczenie początkowych laboratoriów na Programowaniu Obiektowym oraz naukę innych przedmiotów (np. Algorytmów i struktur danych, Metod Numerycznych).

```
#ifndef _WE_WY
```

```
#define _WE_WY
```

zaś ostatnią linią powinna być

```
#endif
```

Powyższe polecenia preprocesora zapobiegają wielokrotnej kompilacji pliku `we_wy.h`, jeśli wielokrotnie jest on włączany do plików tworzących jeden projekt.

Stopniowo, w trakcie kolejnych zajęć, wzbogacaj bibliotekę o funkcje realizujące zadania obsługi wejścia i wyjścia (czytania danych od użytkownika albo wypisywania danych na ekran). Inne funkcje nie powinny tutaj się znajdować.

Zadanie 112.¹³ Napisz funkcję obliczającą sumę wszystkich elementów tablicy.

Zadanie 113.¹⁴ Napisz funkcję znajdującą pozycję, na której znajduje się element minimalny w tablicy liczb całkowitych.

Zadanie 114.¹⁵ Napisz funkcję sortującą tablicę liczb całkowitych. W algorytmie wykorzystaj poprzednią funkcję.

Zadanie 115.¹⁶ Napisz funkcję obliczającą wartość wielomianu w zadanym punkcie.

Zadanie 116. Napisz program 42 z podziałem na funkcje.

Funkcje

Zadanie 102.⁶ Napisz funkcję, która wczytuje jedną liczbę całkowitą. Użytkownik musi mieć możliwość poprawienia błędnie wprowadzonych danych. Zastosuj ją w programie testowym do wczytania trzech liczb.

Zadanie 103.⁷ Napisz funkcję, która wczytuje jedną liczbę całkowitą nieujemną. Użytkownik musi mieć możliwość poprawienia błędnie wprowadzonych danych. Zastosuj ją w programie testowym do wczytania współrzędnych punktu z pierwszej ćwiartki.

Zadanie 104.⁸ Napisz funkcję, która oblicza odległość zadanego punktu od początku układu współrzędnych. W programie testowym wykorzystaj jedną z powyższych funkcji.

Zadanie 105.⁹ Napisz funkcję, która wczytuje do tablicy n liczb całkowitych.

Zadanie 106. Napisz funkcję, która wypisuje n liczb całkowitych umieszczonych w tablicy.

Zadanie 107. Napisz funkcje: wczytującą do tablicy i wypisującą z tablicy n liczb zmiennopozycyjnych. Zastosuj deklarację `typedef`, aby móc łatwo zmieniać typ wczytywanych danych.

Zadanie 108.¹⁰ Napisz funkcję, która przypisuje losowe wartości elementom tablicy liczb całkowitych nieujemnych.

Zadanie 109.¹¹ Napisz funkcję, która przypisuje losowe wartości elementom tablicy liczb zmiennopozycyjnych. Liczby powinny należeć do zadanego przedziału.

Zadanie 110.¹² Napisz funkcję, która prosi o potwierdzenie TAK/NIE i zwraca odpowiednią wartość oznaczającą decyzję użytkownika. Pamiętaj o zapewnieniu możliwości poprawienia błędnie wprowadzonych danych. Tekst ukazujący się na ekranie powinien być argumentem wywołania. Przetestuj działanie funkcji przez trzykrotne jej wywołanie.

Zadanie 111. Umieść wszystkie powyższe funkcje w pliku `we_wy.c`. Stwórz plik nagłówkowy `we_wy.h`, w którym zapisz jedynie nagłówki tych funkcji, każdy zakończony średnikiem. W pliku z rozszerzeniem `main.c` wpisz na początku `#include "we_wy.h"` i umieść funkcję `main` testującą funkcje. Pliki `we_wy.c` i `main.c` włącz do projektu. Skompiluj i uruchom.

Uwaga: Standardowo plik `we_wy.h` powinien zaczynać się

⁶Porównaj program 53.

⁷Porównaj program 53.

⁸Porównaj program 40.

⁹Porównaj program 75.

¹⁰Porównaj program 80.

¹¹Porównaj program 81.

¹²Porównaj program 54.

Pamięć dynamiczna i napisy

Zadanie 117. Napisz funkcję, która rezerwuje pamięć przy pomocy funkcji `malloc` na jednowymiarową tablicę dynamiczną liczb całkowitych i wczytuje do niej dane. Rozmiar tablicy powinien być parametrem funkcji. Zastosuj do utworzonej w ten sposób tablicy wybrane funkcje z zadań 112–115.

Zadanie 118. Napisz funkcję, która rezerwuje pamięć przy pomocy funkcji `malloc` na dwuwymiarową tablicę dynamiczną liczb całkowitych. Rozmiary tablicy powinny być parametrami funkcji. Zastosuj do utworzonej w ten sposób tablicy wybrane funkcje z zadań 91–95.

Zadanie rozwiąż na dwa sposoby:

1. rezerwując oddzielnie pamięć na każdy wiersz,
2. gwarantując, że wszystkie wiersze tablicy zajmą spójny fragment pamięci.

Zadanie 119. Napisz funkcje zwalnijące pamięć zajmowaną przez tablice z poprzedniego zadania.

Zadanie 120. Napisz funkcję, która w zadanym łańcuchu zamienia wszystkie litery wielkie na małe.

Zadanie 121. Napisz funkcję, która sprawdza, czy zadane słowo jest palindromem. Słowem jest tutaj dowolny ciąg liter.

Zadanie 122. Napisz funkcję, która w zadanym łańcuchu usuwa wszystkie wystąpienia zadanego znaku.

¹³Porównaj program 76.

¹⁴Porównaj program 55.

¹⁵Porównaj program 79.

¹⁶Porównaj program 83.

Zadanie 123. Napisz funkcję, która w zadanym łańcuchu zlicza wszystkie wystąpienia zadanego podłańcucha.

Zadanie 124. Napisz funkcję, która w zadanym łańcuchu usuwa wszystkie wystąpienia zadanego podłańcucha. Przetestuj działanie funkcji dla łańcucha `kkotot` i podłańcucha `kot`.

Zadanie 125. Napisz funkcję, która w zadanym łańcuchu każdy podłańcuch będący niezerowym ciągiem spacji zastępuje pojedynczą spacją.

Zadanie 126. Napisz funkcję, która w zadanym łańcuchu usuwa najdłuższy przedrostek nie zawierający cyfry.

Zadanie 127. Napisz funkcję, która wczytuje ciąg słów i umieszcza je w pamięci zarezerwowanej przy pomocy funkcji `malloc`. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100. Nie należy rezerwować więcej pamięci niż potrzeba na przechowanie wczytanych słów.

Zadanie 128. Napisz funkcję, która zwalnia pamięć zajęta przez ciąg słów z poprzedniego zadania.

Zadanie 129. Napisz funkcję, która z zadanego ciągu słów wypisuje słowa, które rozpoczynają się zadaną literą. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100.

Zadanie 130. Napisz funkcję, która z zadanego ciągu słów wypisuje słowa, które kończą się zadaną literą. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100.

Zadanie 131. Napisz funkcję, która w zadanym ciągu słów znajduje słowo, które jest minimalne ze względu na porządek alfabetyczny. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100.

Zadanie 132. Napisz funkcję, która sortuje zadany ciąg słów. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100.

Zadanie 133. Napisz funkcję, która dla zadanego ciągu słów tworzy słowo będące ich złączeniem. Słowem jest tu dowolny ciąg małych liter o długości nie większej niż 100.

Operacje na plikach

Zadanie 134. Napisz program, który otwiera i następnie zamyka plik o podanej przez użytkownika nazwie i w podanym trybie. Sprawdzaj, jakie wartości zwraca funkcja `fopen`. Przed zamknięciem pliku wywołaj funkcje `ferror` i `perror`. Eksperymentuj z różnymi trybami, plikami istniejącymi i nieistniejącymi. Sprawdzaj zawartość pliku po uruchomieniu programu. Zapisz w tabeli wnioski z obserwacji.

Wnioski		
Tryb	Plik istniejący	Plik nieistniejący
"r"		
"w"		
"a"		
"r+"		
"w+"		
"a+"		

Uwaga: należy przeprowadzać testy tylko na takich plikach, których zniszczenie nie spowoduje strat.

Zadanie 135. Napisz program wyświetlający menu i umożliwiający dokonanie następujących czynności:

1. sprawdzenia, czy plik o zadanej nazwie istnieje,
2. zmianę nazwy pliku,
3. usunięcie pliku,
4. skopiowanie pliku tekstowego,
5. wypisanie zawartości pliku tekstowego na ekranie.

Pamiętaj o potwierdzeniu przez użytkownika operacji niebezpiecznych.

Zadanie 136. Napisz funkcję, która zmienia zawartość pliku tekstowego w ten sposób, że każde wystąpienie wybranego znaku, zostaje zastąpione wybranym tekstem.

Zadanie 137. Napisz funkcję, która zmienia zawartość pliku tekstowego w ten sposób, że każdy ciąg białych znaków w pliku zostaje zastąpiony pojedynczym znakiem spacji.

Zadanie 138. Napisz funkcję, która zlicza ile słów występuje w pliku tekstowym. Słowem jest tu każdy ciąg liter otoczony innymi znakami.

Zadanie 139. Napisz funkcję, która wypisuje, w których wierszach pliku tekstowego znajduje się wybrane przez użytkownika słowo. Słowem jest tu każdy ciąg liter otoczony innymi znakami. Program ma wyszukiwać tylko pełne słowa i nie rozróżniać wielkich i małych liter. Można dodatkowo ograniczyć maksymalną długość słowa podanego przez użytkownika.

Zadanie 140. Napisz funkcję, która usuwa wybrane przez użytkownika słowo z pliku tekstowego. Słowem jest tu każdy ciąg liter otoczony innymi znakami, przy czym małe i wielkie litery są rozróżniane. Można dodatkowo ograniczyć maksymalną długość słowa podanego przez użytkownika.

Zadanie 141. Napisz funkcję, która wypisuje największą liczbę zapisaną w zadanym pliku tekstowym. Plik może zawierać tylko liczby całkowite. Jeśli tak nie jest należy wypisać komunikat o błędzie.

Zadanie 142. Napisz funkcję, która zapisuje do pliku tekstowego elementy jednowymiarowej tablicy liczb zmiennopozycyjnych. Format pliku jest następujący: pierwsza liczba musi być całkowitą dodatnią i oznacza liczbę elementów tablicy. Po niej musi znajdować się dwukropek i tyle liczb zmiennopozycyjnych, ile ona wskazuje. Wszystkie liczby oraz dwukropek mogą być oddzielone dowolną liczbą białych znaków. Oto przykładowa zawartość pliku.

```
5 :  
23      13.2      24  
1      -345.2
```

Zadanie 143. Napisz funkcję, która wczytuje z pliku tekstowego liczby zmiennopozycyjne i wstawia je do tablicy. Format pliku jest taki sam, jak w poprzednim zadaniu. Jeśli plik nie jest w powyższym formacie, to powinien zostać wypisany numer wiersza wskazujący na pierwszy błąd.

Zadanie 144. Napisz funkcję, która wczytuje z pliku tekstowego liczby zmiennopozycyjne i wstawia je do tablicy. W pliku znajdują się jedynie elementy tablicy oddzielone dowolną liczbą białych znaków (nie jest na początku podana ich ilość). Tablica powinna zostać zaalokowana przy pomocy funkcji `malloc`. Jeśli plik nie jest w opisanym formacie, to należy podać numer wiersza wskazujący na pierwszy błąd.

Zadanie 145. Napisz funkcję, która zapisuje do pliku elementy dwuwymiarowej tablicy liczb całkowitych. Format pliku musi być następujący. W pierwszym wierszu znajdują się dwie liczby całkowite dodatnie oznaczające odpowiednio liczbę wierszy i kolumn. W następnych liniach zapisane są kolejne wiersze tablicy: w każdej linii dokładnie jeden. Liczby mogą być oddzielone dowolną liczbą spacji i tabulatorów.

Zadanie 146. Napisz funkcję, która wczytuje z pliku tekstowego tabelę liczb całkowitych i wstawia je do tablicy. Format pliku jest taki sam, jak w poprzednim zadaniu. Jeśli plik nie jest w powyższym formacie, to powinien zostać wypisany numer wiersza wskazujący na pierwszy błąd.