

1. Po spotkaniu 14-15 czerwca

Rozwiązania przysyłajcie na p.czarnik@alx.pl

Zadanie 1.1 Pole trójkąta

Napisz program, który odczytuje trzy liczby, sprawdza czy liczby te mogą stanowić boki trójkąta (np. z 2, 2 i 5 nie da się ułożyć trójkąta, prawa?), a jeśli mogą, oblicza pole powierzchni trójkąta o takich bokach.

Wzór Herona: $\sqrt{p(p-a)(p-b)(p-c)}$, gdzie p jest połową obwodu: $(a+b+c)/2$.

Pierwiastek kwadratowy to funkcja `sqrt` z modułu `math`, można też podnieść do potęgi 0.5.

Zadanie 1.2 Naprawa butów / dni tygodnia

Użytkownik podaje, w jaki dzień tygodnia oddaje buty do szewca (numer od 1 do 7), a program losuje czas naprawy butów w dniach, od 0 do 14 dni.

Program ma wypisać, w jaki dzień tygodnia buty będą gotowe do odbioru.

Wypisz dzień odbioru słownie, a jeśli ci się uda, podaj także informację, w jakim tygodniu będzie odbiór, np. „w piątek w tym tygodniu”, „w czwartek za tydzień” albo „w poniedziałek za dwa tygodnie”.

Zadanie 1.3 Firma remontowa

Firma remontowa posiada taki cennik usług:

- gipsowanie ścian: 100 zł za metr kwadratowy ściany
- malowanie ścian i sufitów: 30 zł za metr kwadratowy
- położenie paneli podłogowych: 50 zł za metr kwadratowy podłogi
- położenie listew przypodłogowych: 40 zł za metr bieżący

Napisz program, który pomaga wycenić pracę na podstawie wymiarów pomieszczenia. Zakładając, że pomieszczenie ma kształt prostokąta, program powinien zapytać o dwa wymiary poziome (w metrach) oraz o wysokość i na tej podstawie obliczyć powierzchnię podłogi, sufitu oraz łącznie wszystkich ścian, a także obwód pokoju (listy podłogowe).

Następnie pytaj użytkownika jakie elementy prac mają być wykonane. Na przykład pytanie "Czy gipsować ściany?", a użytkownik odpowiada T lub N, i tak dla każdej pracy.

Na końcu, w zależności od podanych wymiarów pomieszczenia oraz wybranych prac, podawana jest łączna suma do zapłaty.

Zadanie 1.4 Choinka

Napisz program, który wczytuje liczbę całkowitą, a następnie na konsolę wypisuje w tylu liniach „choinkę” ze znaków *. Np. dla parametru 3 powinno się wypisać:

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

Zadanie 1.5 Mniejsze/większe

Program losuje liczbę z zakresu od 1 do 1000. Użytkownik ma zgadnąć tę liczbę nie widząc jej. Kiedy użytkownik poda nieprawidłowy wynik, program podpowiada pisząc czy podana liczba była za duża, czy za mała. Gdy użytkownik poda właściwą liczbę, program wypisuje gratulacje jednocześnie informując, w której próbie udało się zgadnąć liczbę.

Nawiasem mówiąc technika wyszukiwania oparta o „podpowiedzi” za dużo/za mało nazywa się bisekcją i pełni w informatyce bardzo ważną rolę. Umiejętnie ją stosując powinno się te zagadki rozwiązywać w 9-10 próbach (bo $2^{10}=1024$).

Zadanie 1.6 Odgadnij wynik mnożenia

Program losuje dwie liczby i je wypisuje, a użytkownik ma podać wynik mnożenia tych liczb. Jeśli wynik jest poprawny - program "gratuluje"; a jeśli wynik jest niepoprawny - program pyta ponownie tak długo, aż użytkownik w końcu poda prawidłowy wynik. Należy wypisać info, w której próbie udało się odpowiedzieć prawidłowo.

Jako zakres losowania liczby przyjmij co najmniej od 1 do 10, ale może spróbuj z większym zakresem (np. do 20?), aby zagadki były jakimś wyzwaniem...

Przykładowa sesja (możesz dać inne teksty):

Ile to jest 4×6 ?

22

Źle, spróbuj jeszcze raz:

24

Brawo, udało się w 2 próbie!

Zadanie 1.7 Podzielne przez 3 lub przez 5

Użytkownik podaje liczbę całkowitą „limit”. Następnie program wypisuje, bez powtórzeń, liczby z zakresu od 1 do podanego limitu włącznie, które są podzielne przez 3 lub przez 5. Wypisz także jak dużo takich liczb wystąpiło w tym przedziale.

Przykładowe działanie programu:

Podaj limit: 20

3 5 6 9 10 12 15 18 20

Takich liczb było: 9

Zadanie 1.8 Skarb (zadanie z podręcznika)

Napisz grę tekstową polegającą na poszukiwaniu skarbu na dwuwymiarowej planszy o rozmiarach 10 na 10.

Program na początku losuje pozycję skarbu oraz pozycję gracza.

Następnie użytkownik może wprowadzać komendy zmieniające położenie postaci o jedną pozycję w górę/dół/lewo/prawo (np zgodnie z konwencją WSAD) – normalnie za pomocą input.

Gdy gracz wejdzie na pole, na którym kryje się skarb – wygrywa.

Gdy wyjdzie poza planszę – przegrywa.

Po każdym ruchu użytkownik powinien otrzymywać informację o tym, czy zmierza w dobrym kierunku. Po znalezieniu skarbu wypisz liczbę ruchów wykorzystanych przez użytkownika na dojście do celu.