

1. Suma liczb

Napisz program, który prosi o podanie dodatniej niezerowej liczby całkowitej. Program powinien używać pętli do obliczania sumy wszystkich liczb całkowitych z przedziału od 1 do wprowadzonej liczby. Jeśli użytkownik wpisał np. 50, pętla powinna obliczyć sumę ciągu 1, 2, 3, 4, ..., 50.

2. Przebyta odległość

Odległość przebytą przez pojazd można obliczyć w następujący sposób:

$$\text{odległość} = \text{szybkość} \times \text{czas}$$

Na przykład jeśli pociąg jedzie z prędkością 40 km/h przez trzy godziny, to przejechana odległość wynosi 120 kilometrów. Napisz program, który wyświetla prośbę o podanie szybkości pojazdu (w kilometrach na godzinę) i liczbę godzin podróży.

Program powinien wyświetlać w pętli odległość przebytą po każdej godzinie okresu podanego przez użytkownika. Jeśli np. pojazd przez trzy godziny jedzie z szybkością 40 km/h, należy wyświetlić informacje podobne do poniższych:

Godzina Przebyta odległość

1 40

2 80

3 120

Sprawdzanie poprawności danych wejściowych: Program nie powinien akceptować ujemnych prędkości i wartości czasu mniejszych niż 1.

3. Plik z odległością

Zmodyfikuj program z zadania programistycznego 2. („Przebyta odległość”) w taki sposób, aby zapisywał dane w pliku, a nie na ekranie. Otwórz uzyskany plik w Notatniku lub innym edytorze tekstu, aby potwierdzić poprawność danych wyjściowych.

4. Groszowe wypłaty

Napisz program, który oblicza kwotę zarabianą przez pracownika w określonym czasie. Pierwszego dnia pracownik zarabia grosz, drugiego dwa grosze, a każdego następnego — uzyskuje dwukrotność zapłaty za poprzedni dzień. Program powinien wyświetlać tabelę z wynagrodzeniem za każdy dzień, a następnie sumaryczną zapłatę za cały okres. Dane wyjściowe należy wyświetlać w złotych, a nie w groszach.

Sprawdzanie poprawności danych wejściowych: Przy pobieraniu liczby przepracowanych dni nie należy akceptować wartości mniejszych niż 1.

5. Licznik liter

Napisz program, który wyświetla prośbę o podanie łańcucha znaków, a następnie o wprowadzenie znaku. Program powinien zliczać i wyświetlać liczbę wystąpień określonego znaku w łańcuchu.

6. Licznik liter w pliku

Napisz program, który wyświetla prośbę o podanie nazwy pliku, a następnie o wprowadzenie znaku. Program ten powinien zliczać i wyświetlać liczbę wystąpień określonego znaku w pliku. Za pomocą Notatnika lub innego edytora tekstu utwórz prosty plik, który można wykorzystać do przetestowania programu.

7. Obłożenie hotelu

Poziom obłożenia hotelu jest obliczany w następujący sposób:

$$\text{poziom obłożenia} = \text{liczba zajętych pokoi} / \text{łączna liczba pokoi}$$

Napisz program obliczający poziom obłożenia dla każdego piętra hotelu. Program powinien wyświetlać prośbę o podanie liczby pięter, a następnie w pętli wykonywać jedną iterację dla każdego piętra. W każdej iteracji pętla powinna wyświetlać prośbę o podanie liczby wszystkich i zajętych pokoi z danego piętra. Po wszystkich iteracjach program powinien wyświetlić łączną liczbę pokoi w hotelu, liczbę zajętych i pustych pokoi, a także poziom obłożenia.

Sprawdzanie poprawności danych wejściowych: Przy pobieraniu liczby pięter nie należy akceptować wartości mniejszych niż 1. Przy pobieraniu liczby pokoi na piętrze nie należy akceptować wartości mniejszych niż 10.

8. Średni poziom opadów

Napisz program, który za pomocą pętli zagnieżdżonych pobiera dane, a następnie oblicza średni poziom opadów z określonej liczby lat. Program najpierw powinien wyświetlać pytanie o liczbę lat. Pętla zewnętrzna powinna wykonywać jedną iterację dla każdego roku. Pętla wewnętrzna powinna wykonywać 12 iteracji (raz dla każdego miesiąca). W każdej iteracji pętli wewnętrznej należy wyświetlać pytanie o liczbę centymetrów opadów w danym miesiącu. Po wszystkich iteracjach program powinien wyświetlać liczbę miesięcy, łączną liczbę centymetrów opadów i średni poziom opadów

w danym miesiącu z całego okresu.

Sprawdzanie poprawności danych wejściowych: Przy pobieraniu liczby lat nie należy akceptować wartości mniejszych niż 1. Przy pobieraniu miesięcznych opadów nie należy akceptować wartości ujemnych.

9. Populacja organizmów

Napisz program prognozujący wielkość populacji organizmów. Program powinien wyświetlać prośbę o podanie początkowej liczby organizmów, średni dzienny przyrost populacji (w procentach) i liczbę dni, przez jakie organizmy się rozmnażają. Załóżmy, że populacja początkowo liczy dwa organizmy, średni dzienny przyrost wynosi 50%, a czas rozmnażania to siedem dni. Program powinien używać pętli do wyświetlania wielkości populacji każdego dnia.

Sprawdzanie poprawności danych wejściowych: Przy pobieraniu początkowej wielkości populacji nie należy akceptować wartości mniejszych niż 2. Przy pobieraniu średniego dziennego przyrostu populacji należy odrzucać wartości ujemne. Z kolei przy pobieraniu dni rozmnażania nie należy przyjmować wartości mniejszych niż 1.

10. Największa i najmniejsza wartość

Napisz program z pętlą umożliwiającą użytkownikowi podanie serii liczb całkowitych. Użytkownik powinien wpisać -99, aby zasygnalizować koniec serii. Po wpisaniu wszystkich liczb program powinien wyświetlać największą i najmniejszą wprowadzoną wartość.

11. Tabela przeliczająca stopnie Celsjusza na stopnie Fahrenheita

Napisz program, który wyświetla tabelę z temperaturami w stopniach Celsjusza (od 0 do 20) i ich odpowiednikami w stopniach Fahrenheita. Oto wzór na przekształcanie temperatury ze stopni Celsjusza na stopnie Fahrenheita:

W tym wzorze F to temperatura w stopniach Fahrenheita, a C to temperatura w stopniach Celsjusza. Program ma wyświetlać tabelę z temperaturami za pomocą pętli.

12. Wykres słupkowy

Napisz program, który wyświetla prośbę o podanie wartości sprzedaży z danego dnia z pięciu sklepów. Program powinien wyświetlać wykres słupkowy z porównaniem sprzedaży w każdym ze sklepów. Słupki twórz, wyświetlając wiersze gwiazdek. Każda gwiazdka reprezentuje 100 złotych. Oto przykładowe dane wyjściowe programu:

Podaj dzisiejszą wartość sprzedaży dla sklepu nr 1: 1000 [Enter]

Podaj dzisiejszą wartość sprzedaży dla sklepu nr 2: 1200 [Enter]

Podaj dzisiejszą wartość sprzedaży dla sklepu nr 3: 1800 [Enter]

Podaj dzisiejszą wartość sprzedaży dla sklepu nr 4: 800 [Enter]

Podaj dzisiejszą wartość sprzedaży dla sklepu nr 5: 1900 [Enter]

WYKRES SŁUPKOWY Z POZIOMEM SPRZEDAŻY

Sklep nr 1: *****

Sklep nr 2: *****

Sklep nr 3: *****

Sklep nr 4: *****

Sklep nr 5: *****

13. Wyświetlanie początku pliku

Napisz program wyświetlający prośbę o podanie nazwy pliku. Program powinien wyświetlać tylko pięć pierwszych wierszy zawartości pliku. Jeśli plik zawiera mniej niż pięć wierszy, należy wyświetlić zawartość całego pliku.

14. Numery wierszy

Napisz program, który wyświetla prośbę o podanie nazwy pliku. Program powinien wyświetlać zawartość pliku, poprzedzając każdy wiersz jego numerem i dwukropkiem.

Numerowanie wierszy zacznij od 1.

15. Przekształcanie małych liter w pliku na wielkie

Napisz program, który wyświetla prośbę o podanie dwóch plików. Pierwszy plik należy otwierać do odczytu, a drugi do zapisu. Program powinien wczytywać zawartość pierwszego pliku, zmieniać wszystkie znaki na wielkie i zapisywać wynik w drugim pliku.

Drugi plik ma być kopią pierwszego, przy czym wszystkie litery w nim mają być wielkie.

Za pomocą Notatnika lub innego edytora tekstu utwórz prosty plik, który może posłużyć do przetestowania tego programu.

16. Analiza budżetu

Napisz program, który wyświetla użytkownikowi prośbę o podanie miesięcznego budżetu.

Następnie program powinien w pętli pobierać wydatki użytkownika z danego miesiąca i obliczać sumę bieżącą. Po zakończeniu pętli program powinien wyświetlać różnicę między wydaną kwotą a budżetem.

17. Zabawa w odgadywanie liczby losowej

Napisz program, który generuje liczbę losową i wyświetla prośbę o jej odgadnięcie. Jeśli użytkownik poda wartość wyższą od wygenerowanej liczby losowej, program powinien wyświetlać tekst: „Za duża liczba, spróbuj ponownie”. Po podaniu wartości niższej od wygenerowanej program powinien informować: „Za mała liczba, spróbuj ponownie”. W programie należy zastosować pętlę powtarzającą operacje do czasu odgadnięcia liczby losowej przez użytkownika.

18. Rozwinięcie zabawy w odgadywanie liczby losowej

Rozwiń program napisany w zadaniu programistycznym 17., aby zliczał próby odgadnięcia liczby przez użytkownika. Po poprawnym odgadnięciu liczby losowej program powinien wyświetlać liczbę prób.

19. Badanie postrzegania pozazmysłowego

Napisz program, który bada Twoje zdolności pozazmysłowe. Program powinien losowo wybierać jedną z poniższych nazw kolorów:

Czerwony, Zielony, Niebieski, Pomarańczowy, Żółty

Na potrzeby wyboru słowa program może generować liczby losowe. Na przykład liczba 0 może odpowiadać słowu Czerwony, a liczba 1 — słowu Zielony.

Następnie program powinien wyświetlać prośbę o podanie koloru wybranego przez komputer. Po podaniu koloru przez użytkownika program powinien wyświetlać nazwę losowo wybranej barwy. Program powinien powtarzać ten proces 10 razy, a następnie wyświetlać liczbę poprawnych wskazań użytkownika.

20. Wyświetlanie kwadratu

Napisz program, który wyświetla użytkownikowi prośbę o podanie dodatniej liczby całkowitej nie większej niż 15. Program powinien następnie wyświetlać na ekranie kwadrat ze znaków „X”. Liczba podana przez użytkownika określa długość każdego boku kwadratu. Na przykład jeśli użytkownik wpisał 5, program powinien wyświetlić następujący wzór:

XXXXX

XXXXX

XXXXX

XXXXX

XXXXX

Po wpisaniu przez użytkownika 8 program powinien wyświetlać następujące znaki:

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

XXXXXXXX

21. Gra w kości

Napisz program do prostej gry w kości między komputerem a użytkownikiem. Po uruchomieniu program powinien wykonywać pętlę 10 razy. W każdej iteracji pętli należy wykonać następujące operacje:

❑ Wygenerować liczbę losową z przedziału od 1 do 6. Jest to wartość rzutu komputera.

❑ Wygenerować inną liczbę losową z przedziału od 1 do 6. Jest to wartość rzutu użytkownika.

❑ Wyższa liczba oczek daje wygraną. W przypadku równej liczby oczek dla danego rzutu nie ma zwycięzcy.

W kolejnych iteracjach program powinien zachowywać liczbę wygranych komputera i użytkownika. Po wykonaniu wszystkich iteracji program ma wyświetlać, kto wygrał całą grę: komputer czy użytkownik.

22. Symulacja automatu do gier

Automat do gier to maszyna do gier hazardowych, do której użytkownik wrzuca pieniądze, po czym pociąga za wajchę (lub wciska przycisk). Automat do gier wyświetla następnie losowy zbiór obrazków. Jeśli przynajmniej dwa obrazki są takie same, użytkownik wygrywa określoną kwotę pieniędzy wypłacanych przez maszynę.

Utwórz program, który symuluje działanie automatu do gier. Program w trakcie działania powinien wykonywać następujące operacje:

❑ Wyświetlać prośbę o podanie kwoty pieniędzy, jaką użytkownik chce wrzucić do maszyny.

❑ Zamiast wyświetlać obrazki, losowo wybierać słowa z następującej listy:

Wiśnie, Pomarańcze, Śliwki, Dzwonki, Melony, Bary.

❑ Aby wybrać słowo, program może generować liczbę losową z przedziału od 0 do 5.

Po wylosowaniu liczby 0 program wybierze słowo Wiśnie, liczba 1 to Pomarańcze itd. Program powinien trzy razy losowo wybierać słowo z tej listy i wyświetlać każde z tych trzech słów.

❑ Jeśli losowo wybrane słowa nie pasują do siebie, program informuje użytkownika, że wygrał 0 złotych. Jeżeli pasują do siebie dwa słowa, program komunikuje, że użytkownik wygrał dwukrotność postawionej kwoty. Jeśli trzy słowa są identyczne, program wyświetla informację o wygraniu trzykrotności stawki.

❑ Program wyświetla pytanie, czy użytkownik chce zagrać ponownie. Jeśli tak, opisane wcześniej kroki są powtarzane. W przeciwnym razie program wyświetla łączną kwotę wrzuconą do automatu i łączną wartość wygranych.

23. Generator osobistych stron internetowych

Napisz program, który wyświetla prośbę o podanie imienia użytkownika, a następnie o wpisanie opisującego go zdania. Oto przykładowy przebieg programu:

Podaj swoje imię: Julia Tracz [Enter]

Opisz siebie: Studiuję informatykę, należę do zespołu jazzowego, a po studiach chcę pracować jako programistka aplikacji internetowych. [Enter]

Po wprowadzeniu przez użytkownika żądanych danych wejściowych program powinien generować plik HTML z tymi danymi, tworząc w ten sposób prostą stronę internetową.

Oto przykładowy kod w HTML-u oparty na pokazanych wcześniej danych wejściowych:

```
<html>
<head>
</head>
<body>
<center>
<h1>Julia Tracz</h1>
</center>
<hr />
```

Studiuję informatykę, należę do zespołu jazzowego, a po studiach chcę
pracować jako programistka aplikacji internetowych.

<hr />

</body>

</html>