

Programowanie Systemów Komputerowych dla kierunku Mechatronika, sem. 2

Zadania projektowe zaliczeniowe, do samodzielnej realizacji – język C

- Zaliczenie projektów samodzielnych dokonywane jest na podstawie wykonania jednego z programów podanej poniżej listy.
- Projekt zalicza zdobycie min 5 pkt. Maksymalna liczba punktów to 10.
- Jedno zadanie mogą wybrać maksymalnie 4 osoby, ale każdy pracuje samodzielnie, niezależnie od pozostałych osób.
- Jeżeli ktoś ma własny pomysł na zadanie zaliczeniowe, może je wykonać pod warunkiem wcześniejszego zgłoszenia go prowadzącemu w celu ustalenia, czy zadanie ma odpowiedni poziom trudności. Zadania nie zgłoszone wcześniej nie będą przyjmowane.
- Akceptowane są jedynie zadania wykonane w Apache NetBeans 14 w języku C w wersji C99 lub C11. Realizacja w innym środowisku nie jest dopuszczalna.
- Każde z zadań posiada wyszczególnione 3 poziomy trudności. W zależności od wybranego poziomu możliwe jest zdobycie innej maksymalnej liczby punktów:
 - P1 - podstawowy - do 6 pkt.
 - P2 - średniozaawansowany - do 8 pkt.
 - P3 - zaawansowany - do 10 pkt.
- Kryteria oceniania realizacji projektów:
 - Zgodność ze specyfikacją / treścią zadania
 - Kontrola danych wejściowych i obsługa błędów
 - Sposób, algorytm i „elegancja” rozwiązania
 - Styl programowania
 - czytelność kodu (wcięcia, sensowne nazwy zmiennych i funkcji, itp.)
 - podział struktury projektu na oddzielne pliki .c/.h
 - optymalizacja programu
 - komentarze (opis parametrów funkcji, itp.)
 - Styl komunikacji z użytkownikiem (treść i zrozumiałość komunikatów, przejrzysta prezentacja wyników)

Braki w wyżej wymienionych elementach obniżają końcową punktację. Maksymalną liczbę punktów można uzyskać za zadanie oddane w terminie, działające zgodnie z poleceniem / treścią zadania, zawierające czytelny kod źródłowy i wykonane samodzielnie.
- Ocena zadania następuje po przesłaniu kompletnego (wszystkie potrzebne pliki) projektu przez platformę eNauczanie. Prowadzący może zaprosić autora projektu na dodatkowe konsultacje/e-konsultacje w celu omówienia projektu i zadania pytań dot. budowy / działania programu. Konsultacje odbywają się wyłącznie osobiście, w bezpośredniej rozmowie projektu – prowadzący zajęcia.
 - Jeżeli osoba oddająca zadanie nie będzie potrafiła odpowiedzieć na pytania dotyczące swojego programu, zadanie nie zostanie zaliczone.
- Jeżeli treść zadania nie jest jasna, nie jesteś pewna/pewien czy dobrze ją rozumiesz, potrzebujesz jej uściślenia lub podpowiedzi – zapytaj prowadzącego. Lepiej zapytać niż oddać nieprawidłowo zrobione zadanie.
- Termin oddawania zadań – 28 kwietnia 2023r. (ale nie czekajcie do ostatniego dnia! ☺)
 - Oddanie zadania po tym terminie oznacza utratę 0,5 punktu za każdy dzień roboczy zwłoki.
 - Dlaczego opłaca się oddać zadanie przed terminem?:
 - Jeżeli zadanie będzie wymagać poprawek – będzie czas na ich zrobienie i przyniesienie nowej wersji przed terminem końcowym.
 - W przypadku oddawania zadań po terminie, nie będzie możliwości poprawienia – zadanie będzie oceniane w takim stanie, w jakim zostało zaprezentowane, a dodatkowo końcowa liczba punktów będzie obniżana za przekroczenie terminu.

Uwagi dodatkowe:

- Przed oddaniem projektu program należy gruntownie przetestować! Ma nie tylko działać poprawnie dla wszystkich poprawnych danych, ale także być zabezpieczony przed danymi błędnymi i sytuacjami nietypowymi.
- Główny plik kodu źródłowego programu powinien zawierać na początku komentarz z danymi autora i oznaczenie zadania.

- Program ma wyświetlać imię i nazwisko autora oraz numer zadania.
- Program ma wyświetlać informację co użytkownik ma zrobić / wpisać / podać.
- Kod źródłowy programu może być porównany z programami z lat ubiegłych oraz z zasobami internetu w celu eliminacji „gotowców” i kopii. Konsekwencje przyniesienia cudzego programu jako własny mogą być bardzo poważne, z niezaliczeniem przedmiotu łącznie.
 - Wolno korzystać z pomocy i odpowiedzi (np. zapytać na forum internetowym, przedyskutować coś z kolegami), ale nie wolno kopiować cudzych rozwiązań, a zwłaszcza całych programów.
 - Wszelkie wykorzystane, zewnętrzne źródła kodu źródłowego muszą być opisane w pracy (jako komentarz z podaniem źródła, np. adres URL, autora, itp.)

Tematy zadań projektowych

1. Składniki liczby

Poziom P1

Napisz program, który określi, z jakiego zbioru cyfr składa się dowolna zadana liczba, a następnie poda sumę tych cyfr, oraz która z nich jest największa i najmniejsza. Np.: liczba -1246.526 składa się z cyfr 1, 2, 4, 5, 6; suma tych cyfr = 18; najmniejsza cyfra to 1 a największa to 6.

Poziom P2

Program posortuje zbiór cyfr i wyznaczy medianę

2. „Wyciskacz” ciągu liczb

Poziom P1

Napisz program, który wczyta do tablicy ciąg N liczb całkowitych, a następnie usunie z ciągu nadmiar liczb tak, by każda z nich występowała tylko raz. Program powinien podać nowy, zredukowany ciąg oraz sumę wyrazów przed i po redukcji. Liczba N oraz wartości w ciągu podawane będą przez użytkownika. Można założyć, że $N \leq 50$.

Np.: dla ciągu [-2; 0; 2; 0; 1; -9; 4; 5; 5; 5] nowy ciąg to: [-2; 0; 2; 1; -9; 4; 5]. Suma wyrazów przed redukcją = +11, po redukcji = +1.

Poziom P2

Program posortuje zredukowany ciąg malejąco i go wyświetli. Dla ciągu z P1 posortowany ciąg to: [5; 4; 2; 1; 0; -2; -9]. Dodatkowo program poda które liczby i ile razy zostały zredukowane. Np. dla ciągu z P1: liczba 0 - 1 raz, liczba 5 - 2 razy.

3. Korektor tekstu

Poziom P1

Napisz program, który w zadanym ciągu znaków (max 60 znaków):

- Usunie zbędne spacje na końcu i początku tekstu
- Usunie wielokrotne spacje i zamieni je na pojedyncze
- Usunie spacje przed znakami interpunkcyjnymi i nawiasami zamykającymi oraz po nawiasie otwierającym
- Doda spacje po znakach interpunkcyjnych i przed nawiasami otwierającymi, jeśli ich tam nie ma

Poziom P2

Dodatkowo:

- Zamieni litery na początku każdego zdania na wielkie
- Jeśli znajdzie w tekście ciąg „@M” wpisze w to miejsce słowo „mechatronika”

Poziom P3

Umożliwi odczyt z pliku o dowolnej długości zapisze przetworzony tekst do nowego pliku.

4. Operacje na macierzach

Poziom P1

Napisz program, który umożliwi:

- wypełnienie wartościami dwóch tablic / macierzy (rozmiary zadawane przez użytkownika, nie większe niż 10x10; uwaga: tablice nie muszą być kwadratowe).
- dodanie macierzy
- odjęcie macierzy
- mnożenie macierzy

Przed wykonaniem operacji program powinien sprawdzać, czy pozwalają na to wymiary poszczególnych macierzy. Program powinien mieć „menu” umożliwiające wykonanie powyższych operacji oraz zakończenie programu.

Poziom P2

Dodatkowo:

- dodanie do macierzy1 zadanej liczby rzeczywistej
- odjęcie do macierzy1 zadanej liczby rzeczywistej
- mnożenie elementów macierzy1 przez zadaną liczbę rzeczywistą

Poziom P3

Program używa tablic dynamicznych.

5. Przekształcenia macierzy

Poziom P1

Napisz program, który wygeneruje tablicę kwadratową wypełnioną liczbami losowymi, całkowitymi z zakresu $\langle -50, 50 \rangle$ o parzystej liczbie wierszy i o zadanym rozmiarze (max 14×14) podzielny przez 2, a następnie podzieli ją na ćwiartki i przekształci według poniższych zasad:

- Lewą górną – transponuje
- Prawą górną – posortuje w kolejności od lewego górnego, do prawego dolnego rogu, wierszami
- Lewą dolną – zamieni elementy na głównej przekątnej na wartość = 1, elementy na anty-przekątnej na wartość = 0, zaś pozostałym zmieni znak na przeciwny.
- Prawą dolną – obróci o 90 zgodnie z ruchem wskazówek zegara

Pomiędzy poszczególne ćwiartki program powinien wstawić przy wyświetlaniu pusty wiersz i kolumnę.

Np. macierzy wejściową 6×6

1	1	-9	-2	14	2
2	30	2	6	25	-9
-30	0	1	7	-4	2
3	10	4	-26	50	7
4	13	48	34	-23	-18
20	-5	4	1	2	20

Przekształci w macierz:

1	2	-30		-9	-4	-2
1	30	0		2	2	6
-9	2	1		7	14	25
1	-10	0		1	34	-26
-4	1	-48		2	-23	50
0	5	1		20	-18	7

Poziom P2

Program używa tablic dynamicznych a rozmiar macierzy zadawany jest przez użytkownika.

6. Kalkulator długich liczb

Poziom P1

Napisz program umożliwiający sumowanie, dodawanie i mnożenie dowolnie długich liczb całkowitych (tzn. przekraczających zakres typu long double).

Podpowiedź: liczbę należy wprowadzać jako tekst i samodzielnie wykonywać operacje na poszczególnych cyfrach, pamiętając o przeniesieniach.

Poziom P3

Dodatkowo program umożliwi także dzielenie liczb.

7. Statystyka tekstu

Poziom P1

Napisz program, który w podanym z klawiatury, dowolnym tekście (zdaniu) policzy ile w tym tekście jest:

- znaków (razem: liter, spacji, liczb, innych znaków);
- spacji;
- liter (wszystkich);
- dużych liter;

Poziom P2

Dodatkowo

- cyfr;
- liczb;
- słów;

Poziom P3

Wykona obliczenia dla tekstu odczytanego z pliku tekstowego.

8. Szukanie ciągu rosnącego

Poziom P1

Napisz program, który wczyta do tablicy ciąg N liczb rzeczywistych, a następnie znajdzie najdłuższy podciąg liczb rosnących. Program powinien podać: pozycję pierwszego wyrazu znalezionej podciągu,

wyrazy znalezionej podciągu, ich liczbę i sumę. Liczba N oraz wartości w ciągu podawane będą przez użytkownika. Można założyć, że $N \leq 25$.

Np.: dla ciągu $[-2.5; 0; -3.1; 2; 1; -1.1; 3.4; 5; 4.9; -5]$ szukany podciąg to $[-1.1; 3.4; 5]$ i ma on 3 wyrazy, a ich suma to 7.3, zaś podciąg zaczyna się od 6-go wyrazu ciągu wejściowego.

Poziom P2

Jak wyżej, ale bez ograniczenia długości ciągu do 25.

9. Badanie ciągu liczb

Poziom P1

Napisz program, który wczyta dla zadanego ciągu N liczb rzeczywistych sprawdzi, czy jest to ciąg rosnący, niemalejący, stały, nierosnący, malejący czy niemonotoniczny. Liczba N oraz wartości w ciągu podawane będą przez użytkownika.

Poziom P2

Program powinien także obliczyć wartość średnią ciągu i odchylenie standardowe. W przypadku użycia tablic, muszą to być tablice dynamiczne.

Poziom P3

Program pozwoli wykonać obliczenia także dla ciągu liczb odczytanego z pliku tekstowego (format: 1 wiersz – 1 liczba).

10. Szukanie ciągu maksymalnego

Poziom P1

Napisz program, który wczyta do tablicy ciąg N liczb rzeczywistych, a następnie znajdzie podciąg o największej sumie liczb. Program powinien podać: pozycję pierwszego wyrazu znalezionej podciągu, wyrazy znalezionej podciągu, ich liczbę i sumę. Liczba N oraz wartości w ciągu podawane będą przez użytkownika. Można założyć, że $N \leq 25$.

Np.: dla ciągu $[-2.5; 0; -3.1; 2; 1; -1.1; 3.4; 5; 4.9; -5]$ szukany podciąg to $[2; 1; -1.1; 3.4; 5; 4.9]$ i ma on 6 wyrazów, a ich suma to 15.2, zaś podciąg zaczyna się od 4-go wyrazu ciągu wejściowego.

Poziom P2

Liczba N może być dowolna, a tablica jest tablicą dynamiczną.

Poziom P3

Program pozwoli wykonać obliczenia także dla ciągu liczb odczytanego z pliku tekstowego (format: 1 wiersz – 1 liczba).

11. Kalkulator ułamków

Poziom P2

Napisz program, który wykona dodawanie, odejmowanie, mnożenie i dzielenie dwóch ułamków wymiernych, podanych w postaci a/b . Wynik powinien być sprowadzony do najprostszej postaci.

Np.:

$$3/4 + 4/5 = 1 + 11/20$$

$$3/4 - 4/5 = -1/20$$

$$3/4 * 4/5 = 3/5$$

$$3/4 / 4/5 = 15/16$$

12. Tłumacz dat

Poziom P2

Napisz program sprawdzający poprawność podanej daty oraz podający tekstową postać tej daty, np.: 1910 07 15 = „piętnasty lipca tysiąc dziewięćset dziesiątego roku”.

Uwagi pomocnicze:

Kalendarz gregoriański obowiązuje od roku 1582r.

Lata przestępne to te, których numer jest podzielny przez 4 z wyjątkiem lat wyrażających się setkami.

Lata będące wielokrotnością 400 są jednak latami przestępnymi.

13. Tłumacz liczb

Poziom P2

Napisz program zamieniający podaną liczbę na tekst, np.: -1234567.89 = „minus jeden milion dwieście trzydzieści cztery tysiące pięćset sześćdziesiąt siedem i osiemdziesiąt dziewięć setnych”. Zakres liczb: -/+999'999'999,999

14. Skośne w macierzy

Poziom P1

Napisz program, który zbuduje i wyświetli tablicę prostokątną $m \times n$ (max 15×10) liczb losowych całkowitych z przedziału $[-50, 50]$, a następnie obliczy sumę liczb znajdujących się na skośnych przechodzących przez element o zadanych współrzędnych (w, k) oraz różnicę między elementem maksymalnym i minimalnym z obu skośnych

Np. dla $m=4$ i $n=6$ oraz $w=2$ i $k=3$

		k			n	
w	50	-10	6	45	3	29
	33	3	16	0	-2	-1
	-4	-19	-21	2	-7	-12
m	22	36	-17	4	-46	39

Suma = 10

Różnica max-min=91

Poziom P2

Program używa tablic dynamicznych.

Poziom P3

Program pozwoli wykonać obliczenia także dla tablicy liczb odczytanej z pliku tekstowego.

15. Maksima macierzy

Poziom P1

Napisz program, który wygeneruje tablicę o zadanej rozmiarze (max 15×10) liczb losowych, a następnie znajdzie pozycje elementów, których wartości są większe od wartości wszystkich elementów sąsiednich. Wyświetl tablicę i podaj ich położenie.

Np. w macierzy 4×6

1	1	-9	-2	1	2
2	3	2	6	5	-9
-3	0	1	7	-4	2
2	-5	4	1	2	2

Szukane elementy (maksima lokalne): to 3 – poz. 2,2 | 7 – poz. 3,4 | 2 – poz. 4,1

Poziom P2

Program używa tablic dynamicznych.

Poziom P3

Program pozwoli wykonać obliczenia także dla tablicy liczb odczytanej z pliku tekstowego.

16. Szukanie bloku dodatniego

Poziom P1

Napisz program, który wygeneruje tablicę prostokątną o zadanej rozmiarze (max 15×10) liczb losowych, a następnie znajdzie, w tablicy największy (co do liczby elementów) blok liczb dodatnich. Program powinien podać rozmiar bloku oraz pozycję jego pierwszego i ostatniego elementu.

Np. w macierzy 4×6

1	1	-9	-2	1	2
2	3	2	6	5	-9
-3	0	1	7	-4	2
2	-5	4	1	2	2

Znaleziono blok 3×2 , Pozycja początku bloku = 2, 3, pozycja końca bloku = 3, 4

Poziom P2

Program używa tablic dynamicznych.

Poziom P3

Program pozwoli wykonać obliczenia także dla tablicy liczb odczytanej z pliku tekstowego.

17. Edytor tekstu

Poziom P2

Napisz program umożliwiający edycję kilku linii tekstu (np. pięciu). Program powinien dawać możliwość: wpisania tekstu, przesuwania kursora we wszystkich kierunkach i wstawienia lub usunięcia znaków w dowolnym miejscu tekstu, obsługi klawiszy Delete, Backspace, Insert, Home, End, Enter. Maksymalna długość tekstu w jednej linii to 60 znaków.

Poziom P3

Dodatkowo program ma umożliwiać zapis i odczyt tekstu do / z pliku.

18. Pole kółek

Poziom P2

Napisz program, który na podstawie współrzędnych środków N kół na płaszczyźnie i ich promieni obliczy całkowitą powierzchnię powstałej wypadkowej figury złożonej z wszystkich kół. Uwaga: koła mogą się częściowo lub całkowicie pokrywać!

Wraz z zadaniem należy przynieść kartkę z min. 3 obliczonymi przykładami w celu sprawdzenia poprawności działania programu

19. Hacker

Poziom P3

Dostęp do serwisów bankowych często zabezpieczony jest kodem PIN (np. 8 cyfrowym). Podczas logowania system pyta o losowo wybrane 3 cyfry. Napisz program, który spróbuje odtworzyć nieznaną PIN na podstawie listy 5 kombinacji, które zapewniły poprawne logowanie. Jeśli nie da się odtworzyć PIN – program powinien podać liczbę możliwych pozostałych kombinacji oraz zapisać je do pliku.

Ogólny algorytm programu:

- Generowanie danych
 - Podanie przez użytkownika znanego PIN
 - Wygenerowanie 5 prób logowania (losowe wybranie 3 cyfr + informacja które to cyfry) i zapis ich do pliku tekstowego
- Poszukiwanie
 - Odczytanie pliku z próbami logowania i próba odtworzenia PIN
- Wyniki
 - Podanie znalezionego PIN (i zapisanie go do pliku tekstowego) lub podanie liczby możliwych kombinacji (i zapisanie ich wszystkich do pliku tekstowego)

20. Interpolator

Poziom P3

Napisz program, który odczytując z pliku tekstowego pewien zbiór danych (2 kolumny wartości X i Y , wartości X wyłącznie rosnące, nie koniecznie w równych odstępach, liczby zmiennopozycyjne) wykona interpolację danych dla nowych punktów, odległych od siebie o zadaną przez użytkownika wartość. Nowe wyniki mają zostać zapisane do nowego pliku tekstowego. Interpolacja ma zostać wykonana funkcją liniową. Wyniki zapisać do pliku i napisać dodatkowy skrypt do Matlaba pokazujący wykres.

Np.: dla danych

X	Y
0	1
2	0
3	2

i nowej podziałce = 0.5, wyniki powinny być:

X	Y
0	1
0,5	0,75
1	0,5
1,5	0,25
2	0
2,5	1
3	2

21. Tester czworokątów

Poziom P2

Napisz program, który wczyta współrzędne 4 wierzchołków czworokąta, i zbada, jaki czworokąt opisują te wierzchołki (kwadrat, prostokąt, równoległobok, trapez, romb, czworokąt wypukły czy wklęsły).

Wraz z zadaniem należy przynieść kartkę z min. 10 obliczonymi przykładami w celu sprawdzenia poprawności działania programu.

22. Grzebień do plików

Poziom P2

Napisz program, który odczyta plik tekstowy ASCII i rozdzieli go na dwa pliki w taki sposób by w każdym z nich znalazł się co drugi znak. Uwaga, znaki nowej linii mają zostać zachowane w obu plikach. Np.:

Plik wejściowy	Plik wyjściowy 1	Plik wyjściowy 2
Ala ma kota. Kot Ali ma ogon i 4 lapy. Ala miała też myszki, ale kot je zjadł.	Aam oa o l aoo ay l il e yzi l o ezal	l akt. KtAim gni4lp. Aamaatzmsk,aektj jd.

Poziom P3

Dodatkowo program da wybór czy plik ma być dzielony co drugi znak czy co drugie słowo. Przykład dla drugiego przypadku:

Plik wejściowy	Plik wyjściowy 1	Plik wyjściowy 2
Ala ma kota. Kot Ali ma ogon i 4 lapy. Ala miała też myszki, ale kot je zjadł.	Ala kota. Ali ogon 4 Ala też ale je	ma Kot ma i lapy. miała myszki, kot zjadł.

23. Sklejacz plików

Poziom P2

Napisz program o działaniu odwrotnym do programu z zadania 22 – poziom 2, tzn. łączący dwa pliki wejściowe w jeden wyjściowy tak by odtworzyć pierwotny tekst.

Poziom P3

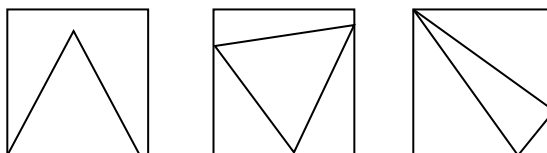
Napisz program o działaniu odwrotnym do programu z zadania 22 – poziom 3, tzn. łączący dwa pliki wejściowe w jeden wyjściowy tak by odtworzyć pierwotny tekst.

24. Trójkąt w kwadracie

Poziom P2

Napisz program, który wczyta długość boku kwadratu oraz trzy długości boków trójkąta a następnie stwierdzi, czy podany trójkąt da się umieścić wewnątrz zadanego kwadratu (uwzględniając także możliwość obrócenia trójkąta).

Wraz z zadaniem należy przynieść kartkę z min. 5 obliczonymi przykładami w celu sprawdzenia poprawności działania programu.



Poziom P3

Program ma podawać, jaki jest potrzebny kąt obrotu trójkąta zakładając, że pierwszy z jego podanych boków był przed obrotem równoległy do dolnego boku kwadratu.

25. Koder UART

Poziom P2

Napisz program, który zadany ciąg znaków przekształci na ciąg binarny zgodnie z zasadami transmisji dla typowego UART. Użytkownik powinien mieć możliwość wyboru długości ramki (5...8 bitów), obecności bitu parzystości i liczby bitów stopu (1 lub 2). Stan wysoki ma być kodowany jako „1”, a niski jako „0” (uwaga: odwrotnie niż w RS232)

Np. dla ciągu znaków „1A”, ramka 8 bit, kontrola parzystości, 1 bit stopu, program powinien wygenerować ciąg „10011000111 10100000101”

Więcej o UART i kodach ASCII, np.:

<https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html>

<https://www.circuitbasics.com/basics-uart-communication/>

<https://pl.wikipedia.org/wiki/ASCII>

Poziom P3

Program wykonuje powyższe przekształcenia dla danych tekstowych odczytanych z pliku. Przekształcony ciąg zapisywany jest do innego pliku.

26. Dekoder UART

Poziom P2

Napisz program, który zadany ciąg bitów zakodowanych zgodnie z zasadami transmisji dla typowego UART przekształci w ciąg znaków ASCII. Działanie programu ma być odwrotne do programu z zadania 25.

Poziom P3

Program wykonuje powyższe przekształcenia dla danych tekstowych odczytanych z pliku. Przekształcony ciąg zapisywany jest do innego pliku.