**9. Підпрограми**

**6.1.** Скласти програму обчислення добутку

*p=f0**f1*...*fn,*

де *fl*=http://obvintsev.info/compuscience/problems/Z_Theme6_files/image002.gif

**6.2.** Два пpостих числа називаються "близнюками", якщо вони відpізняються один від одного на 2 (напpиклад, числа 41та 43). Скласти програму виведення на друк всіх паp "близнюків" з відpізку [*n,*2*n*], де *n* - задане ціле число, яке більше 2.

**6.3.** Дано натуральне число *n* та послідовність натуральних чисел *a1, a2, …, an*. Показати всі елементи послідовності, які є

а) повними квадратами;

б) степенями п’ятірки;

в) простими числами.

Визначити відповідні функції для перевірки, чи є число: повним квадратом, степенню п’ятірки, простим числом.

**6.4.** Дано натуральне число *n*. Для чисел від 1 до *n* визначити всі такі, які можна представити у вигляді суми двох повних квадратів. Описати функцію, яка перевіряє, чи є число повним квадратом.

**6.5.** Дано парне число *n*>2. Перевірити для нього гіпотезу Гольдбаха, яка полягає в тому, що кожне парне число *n*>2 можна представити у вигляді суми двох простих чисел. Визначити функцію, яка перевіряє, чи є число простим.

**6.6.** Скласти алгоритм обчислення величини

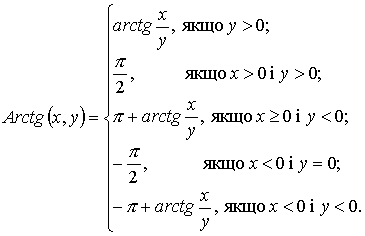
http://obvintsev.info/compuscience/problems/Z_Theme6_files/image004.gif

для заданого дiйсного числа *a>*0. Визначити функцiю обчислення коренiв http://obvintsev.info/compuscience/problems/Z_Theme6_files/image006.gif з точнiстю  за наступною iтерацiйною схемою

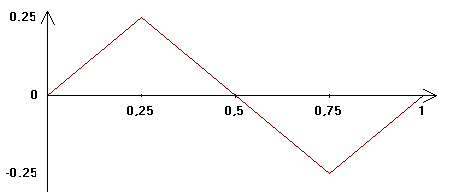
http://obvintsev.info/compuscience/problems/Z_Theme6_files/image008.gif,

взявши за вiдповiдь наближення http://obvintsev.info/compuscience/problems/Z_Theme6_files/image010.gif, для якого http://obvintsev.info/compuscience/problems/Z_Theme6_files/image012.gif.

**6.7.** Використовуючи функцiю *y=arctg*(*x*) (*math.atan(x)*), скласти пiдпрограму для обчислення функцiї, заданої спiввiдношенням



**6.8.** Скласти програму обчислення значень функцiї *f(x)*, перiодичної з перiодом 1 i визначеної на всiй числовiй вісi. Графiк функцiї зображено на малюнку 6.1



Мал. 6.1 - Графік періодичної функції до завдання 6.7.

Якi допомiжнi пiдпрограми будуть потрiбнi для розв'язку задачi ?

**6.9.** Визначити функцiю для обчислення елiптичного iнтегралу

http://obvintsev.info/compuscience/problems/Z_Theme6_files/image017.gif

який, як показав Гаусс , рiвний границi http://obvintsev.info/compuscience/problems/Z_Theme6_files/image019.gif монотонно-збiжних послiдовностей *an* i *bn*, які визначаються рекурентними спiв­вiдношеннями

http://obvintsev.info/compuscience/problems/Z_Theme6_files/image021.gif

Вказана границя називається арифметико-геометричним середнiм чисел *a* i *b*.

*Вказiвка.*При виборi умови повторення циклу врахувати , що

http://obvintsev.info/compuscience/problems/Z_Theme6_files/image023.gif

**6.10.** Визначити функції для обчислення

а) синуса ; б) косинуса

використовуючи їх розклади в ряд Тейлора.

**6.11.** Дано координати вершин трикутника і точки всередині його. Використовуючи функцію для обчислення площі трикутника через три його сторони, визначити відстань від даної точки до найближчої сторони трикутника.

*Вказівка.*Врахувати що площа трикутника обчислюється також через основу і висоту.

**6.12.** Скласти функцію перевірки заданого рядка на симетричність.

**6.13.** Перевірити, чи є даний рядок ідентифікатором, натуральним числом, чи ні тим ні іншим. Скласти функції, які визначають чи є заданий символ літерою та чи є даний символ цифрою.

**6.14.** Скласти функцію, яка визначає позицію першого (останнього) входження заданого символа в заданий рядок.

**6.15.** Cкласти процедуру, яка замiнює в початковому рядку символiв всi одиницi на нулi, а всi нулi - на одиницi. Замiна повинна виконуватись, починаючи з заданої позицiї рядка.

**6.16.** Скласти процедуру, в результатi звернення до якої з першого заданого рядка видаляється кожний символ, який належить i другому заданому рядку.

**6.17.** Скласти підпрограму для обчислення значення натурального числа за заданим рядком символів, який є записом цього числа у системі числення за основою *b*(2<*b*<16). Використати функцію, яка за заданим символом повертає відповідну цифру у системі числення за основою *b*.

**6.18.** Скласти підпрограму для отримання за заданим натуральним числом рядка символів, який є записом цього числа у системі числення за основою *b* (2<*b*<16).Використати функцію, яка за заданою цифрою у системі числення за основою *b* повертає символ, що відповідає цій цифрі.

**6.19.** Скласти алгоритм додавання „у стовпчик” двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

1) функцію *GetDigit(c)* отримання цифри за символом *c*;

2) функцію *GetSymbol(d)* отримання символа за цифрою *d*;

3) процедуру *AddDigit(n1, n2, p, n)* додавання двох цифр *n1*, *n2* з урахуванням перенесення p та отримманням останньої цифри результату *n*;

4) функцію додавання двох рядків у стовпчик *AddColumn*(*S1*, *S2*).

**6.20.** Скласти алгоритм множення „у стовпчик” двох чисел, записаних у вигляді рядків, що є позиційними записами цих чисел у десятковій системі числення. Використати підпрограми:

1) функцію *GetDigit(c)* отримання цифри за символом *c*;

2) функцію *GetSymbol(d)* отримання символа за цифрою *d*;

3) процедуру *MulDigit(n1, n2, p, n)* множення двох цифр *n1*, *n2* з урахуванням перенесення *p* та отримманням останньої цифри результату *n*;

4) підпрограму *MulStrChar*(*S*, *c*) множення рядка *S* на символ *c*;

5) підпрограму *AddString*(*S1*, *S2*, *n*) додавання двох рядків у стовпчик зі „зсувом” другого рядка на *n* позицій ліворуч.

**6.21.** Скласти процедуру "стискання" рядка: кожний пiдрядок, який складається з кiлькох входжень одного i того ж символа, замiнюється самим цим символом.

**6.22.** Скласти підпрограми для

а) підрахунку кількості слів рядка;

б) отримання найдовшого слова;

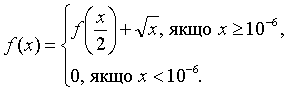
в) отримання найкоротшого слова;

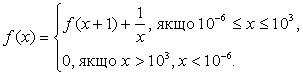
г) отримання всіх слів, які є паліндромами (симетричними);

д) отримання всіх слів, які є ідентифікаторами;

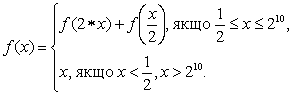
д) отримання всіх слів, які є натуральними числами.

**6.23.** Скласти рекурсивнi пiдпрограми для обчислень значень функцiй

а) 

б) 

в) http://obvintsev.info/compuscience/problems/Z_Theme6_files/image029.gif

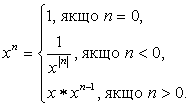
г) 

**6.24.** Скласти рекурсивну функцiю для обчислення многочленiв Ермiта (див. завдання 3.15 б) з теми 3.2. „Програмування рекурентних співвідношень”) i порiвняти кількість дій у рекурсивному та нерекурсивному варiантах.

**6.25.** Визначити рекурсивну функцiю обчислення *НСД*(*n,m*) натуральних чисел, яка грунтується на спiввiдношеннi *НСД*(*n,m*)=*НСД*(*m,r*), де *r* - остача вiд дiлення *n* на *m*.

**6.26.** Визначити рекурсивну процедуру представлення натурального числа *Z* у вiсiмковiй системi числення.

**6.27.** Визначити рекурсивну функцiю обчислення степеня дiйсного числа з цiлим показником *xn* згiдно з формулою



**6.28.** Визначити рекурсивну функцiю для обчислення бiномiального коефiцiенту http://obvintsev.info/compuscience/problems/Z_Theme6_files/image035.gif http://obvintsev.info/compuscience/problems/Z_Theme6_files/image037.gif, за такою формулою:

http://obvintsev.info/compuscience/problems/Z_Theme6_files/image039.gif, при 0<*m<n*.

**6.29.** Визначити рекурсивну функцiю для знаходження суми додатнiх дiйсних чисел, якi складають непорожню послiдовнiсть, за якою слiдує вiд'ємне число.

**6.30.** Визначити рекурсивну функцiю для обчислення числа Фiбоначчi *Fn* для заданого натурального *n* (див. завдання 10 з теми “Арифметичний цикл”). Порiвняти працемісткість рекурсивного i нерекурсивного варiантiв.

**6.31.** Задані натуральнi числа *a,c,m*. Визначити рекурсивну функцiю для обчислення *f*(*m*) за формулою

http://obvintsev.info/compuscience/problems/Z_Theme6_files/image041.gif,

*g*(*m*) - остача вiд дiлення *an*+*c* на 10.

**6.32.** Визначити рекурсивнi функцiї

а) перевiрки заданого рядка на симетричнiсть;

б) побудови рядка, iнвертованого по вiдношеню до заданого;

в) замiни у вихiдному рядку всiх входжень даного символа даним рядком;

г) перевiрки, чи є один рядок початком iншого;

д) перевiрки на входження одного рядка у iнший.

*Вказiвка*. Нехай *, А, В*  *W* ( - порожній рядок), *х,у*  *Ch*.

Для побудови рекурсивних функцiй використати спiввiдношення

а) *сим*() = *Іст,* *сим*(*add*(*х,* ) = *Іст,*

*сим*(*app*(*add*(*х, А*)*, у*)) = (*х=у*) & *сим*(*А*);

б) *інв*() = ,

*інв*(*add*(*х, А*) = *app*(*інв*(*А*), *х*);

в) *зам*(*, х, В*) = ,

*зам*(*add*(*у, А*)*, х, В*) = *add*(*у, зам*(*А, х, В*)),

*зам*(add(*х, А*)*, х, В*) = *В+зам*(*А, х, В*);

г) *поч*(*, В*) = *Іст*, *поч*(*add*(*x*, *А*)*,* ) = *Хиб*,

*поч*(*add*(*х*, *А*), *add*(*у*, *В*)) = (*х = у*) & *поч*(*А, В*);

д) *вход*(*, В*) = *Іст*, *вход*(add(*х*, *А*),) = *Хиб*,

*вход*(*add*(*х*, *А*), *add*(*у*, *В*)) = *поч*(*add*(*х*, *А*), *add*(*у*, *В*))*вход*(add(*х*, *А*)*, В*).

**6.33.** Скласти рекурсивну функцiю для обчисленя функцiї Аккермана *Акк*(*n,m*), заданої спiввiдношенням

*Акк*(*0,m*)*=m+1;*

*Акк*(*n,0*)=*Акк*(*n-1,1*)*;*

*Акк*(*n,m*)*=Акк*(*n-1,Акк*(*n,m-1*)).

Обчислити *Акк*(0,5),*Акк*(1,2),*Акк*(2,2).

Покажемо спосiб обчислення функцiї Аккермана на прикладi:

*Акк*(1,2)=*Акк*(0,*Акк*(1,1))=*Акк*(0,*Акк*(0,*Акк*(1,0)))=  
*Акк*(0,*Акк*(0,*Акк*(0,1)))=*Акк*(0,*Акк*(0,2))=*Акк*(0,3)=4.

**6.34.** Скласти рекурсивну функцію обчислення суми:



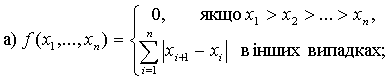
**6.35.** Ханойськi вежі. Дошка має три стрижні. На перший нанизано *N* дискiв спадного догори дiаметра. Потрiбно, перекладаючи диски по одному, розмiстити їх в початковому порядку на другому стрижнi. При цьому бiльший диск нiколи не повинен розмiщуватись над меншим.

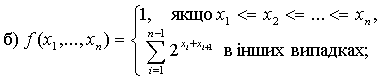
Скласти пiдпрограму, яка iлюструє порядок перемiщення дискiв. Викликати її при *N*=3. Підрахувати кiлькiсть ходiв, які потрiбнi для перемiщен­ня дискiв. Знайти її залежнiсть вiд *N*.

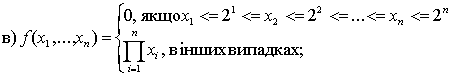
**6.36.** Скласти програму, яка вiдображає всi перестановки цiлих чисел вiд 1 до *N*.

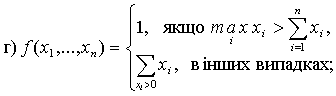
*Вказiвка*. Множину перестановок цiлих чисел вiд 1 до *N* можна отримати з множини всiх перестановок цiлих чисел вiд 1 до *N*-1, вставляючи *N* в усi можливi позицiї в кожнiй перестановцi.

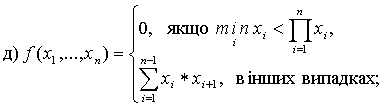
**7.40.** Скласти підпрограми зі змінною кількістю параметрів для обчислення функцій











http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image014.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image016.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image018.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image020.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image022.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image024.gif

*Вказівка*: оформити *xi* як позиційні, а *yi*, - як ключові параметри.

Додати задачі для змінної кількості параметрів та для ключових параметрів а також для lambda-функцій.

**7.166.** Визначити функцiю для обчислення кореня рiвняння *f*(*x*)*=*0 на вiдрiзку [*a,b*], на якому *f*(*x*) змінює знак, з заданою точнiстю  методом дiлення вiдрiзка навпіл. Виконати обчислення кореня для функції *f*(*x*)=*x3-7\*x-1*.

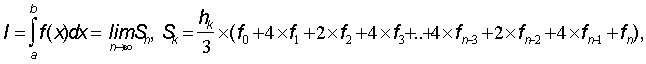
**7.167.** Визначити функцiю для обчислення кореня рiвняння *f*(*x*)*=*0 на вiдрiзку [*a,* *b*] , на якому *f*(*x*) змінює знак, з заданою точнiстю  методом хорд. Виконати обчислення кореня для функції *f*(*x*)=*x3-7\*x-1*.

*Вказiвка*. Обчислюючи корінь методом хорд, з’єднують прямою точки (*a*, *f*(*a*)) та (*b*, *f*(*b*)) та знаходять точку *x* перетину цієї прямої з віссю абсцис. Якщо знаки *f*(*a*) та *f*(*x*) співпадають, далі пошук проводять на відрізку [*x*, *b*], інакше – на відрізку [*a*, *x*].

**7.168.** Скласти пiдпрограми для обчислення визначеного iнтегралу

а) методом прямокутників; б) методом Сiмпсона.

*Вказiвка* б).Для обчислення iнтегралу використати границю



де http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image006.gif i представлення http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image008.gif у виглядi

http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image010.gif

де

http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image012.gif

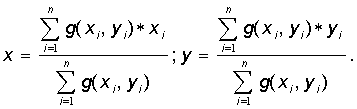
Для обчислення http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image014.gif використати рекурентне співвідношення

http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image016.gif

http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image018.gif

http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image020.gif

**7.169.** Нехай *М1,М2,...,Мn* - матеріальні точки, положення яких на площині задано координатами *(x1,y1),...,(xn,yn)*, а маси визначаються за допомогою вагової функції*g(x,y)*. Положення центру ваги цих точок задано формулами:



Визначити функцію обчислення точки центру ваги (*х, у*) при заданій ваговій функції *g*(*x,y*).

Складену функцію використати для знаходження положення центра ваги *n* точок при http://obvintsev.info/compuscience/problems/Z_Theme7_3_files/image024.gif

**7.170.** Скласти підпрограму для знаходження елемента дійсного вектора, який задовольняє умову, задану булівською функцією *Q*(*x*). Виконати пошук, коли умовою *Q*(*x*)є:

а) *x*=*a*; б) *x*>*a*; в) *a*<=*x*<=*b*;

де *a*, *b* – задані числа.

**T9.1** Скласти функцію для обчислення скалярного добутку двох векторів з використанням lambda-функції для обчислення добутку двох чисел.

*Вказівка*: використати функцію sum()

**T9.2** Скласти функцію для обчислення добутку матриці розміром mxn на вектор розміром n з використанням lambda-функції.

*Вказівка*: використати функцію обчислення скалярного добутку двох векторів та спискоутворення.

**T9.3** Скласти функцію для обчислення добутку вектору розміром m на матрицю розміром mxn з використанням lambda-функції.

*Вказівка*: використати функцію обчислення скалярного добутку двох векторів та спискоутворення.

**T9.4** Скласти функцію для обчислення добутку матриці розміром mxn на матрицю розміром nxk з використанням lambda-функції.

*Вказівка*: використати функцію обчислення скалярного добутку двох векторів та спискоутворення.

**T9.5** Скласти функцію для обчислення суми двох матриць розміром mxn з використанням lambda-функції.

*Вказівка*: використати спискоутворення.

**T9.6** Скласти функції обчислення норм дійсної матриці порядку *n* з використанням lambda-функції.

а)http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image036.gif, б) http://obvintsev.info/compuscience/problems/Z_Theme7_2_files/image038.gif

**T9.7** Скласти функцію, що перевіряє чи є задана цілочисельна квадратна матриця магічним квадратом, тобто такою, в якій суми елементів в усіх рядках і стовпчиках однакові. Використати lambda-функцію.

*Вказівка*: використати функцію zip() для отримання транспонованої матриці.