

Timpul acordat: 180 min
„ASPIRAȚII INFORMATICE” – 2018
CLASELE XI – XII

1. GHICEȘTE NUMĂRUL

Vlad ascunde un număr k cuprins între 1 și n . Calculatorul încearcă să-l ghicească, evident efectuând cât mai puține verificări. Fiind necunoscut numărul k , calculatorul alege de fiecare dată numărul de la mijlocul segmentului curent (cifrele zecimale se anulează). Dacă:

- numărul ales de calculator este mai mare decât k , atunci se generează caracterul M ;
- numărul ales de calculator este mai mic decât k , atunci se generează caracterul P ;
- numărul ales de calculator este egal cu k , atunci se generează caracterul E și algoritmul se încheie.

Sarcină. Elaborați un program, care să simuleze un astfel de joc.

Date de intrare. Fișierul text GHICI.IN conține pe prima linie 2 numere naturale n și k – care reprezintă limita superioară pentru numere, și, respectiv, numărul, ascuns de Vlad.

Date de ieșire. Fișierul text GHICI.OUT va conține pe prima linie setul de caractere, generat de calculator.

Restricții:

- $1 \leq n \leq 1000$; $1 \leq k \leq n$;
- Timpul de execuție nu va depăși 1 s;
- Programul va folosi cel mult 32 MB de memorie operativă.

Exemplu:

GHICI.IN	GHICI.OUT	Explicații
100 78	PPMME	Calculatorul va genera numerele 50, 75, 87, 81, 78. Respectiv caracterele generate P (50<78), P (75 < 78) , M (87>78) , M (81>78), E (78=78).

2. TELEFON

Numerele de telefon din agenda telefonică a lui Ionel au unul din următoarele formate: $+7 <cod> <număr>$; $8<cod> <număr>$; $<număr>$, unde $<număr>$ este format din 7 cifre, iar $<cod>$ este format din 3 cifre sau trei cifre în paranteze rotunde. Dacă codul nu este specificat, atunci se presupune că acesta este egal cu 373. De asemenea, în numerele de telefon, poate apare semnul „,” între oarecare două cifre. Momentan, agenda telefonică a lui Ionel conține n numere înregistrate și el dorește să mai înregistreze un număr de telefon.

Sarcină. Elaborați un program, care să determine dacă numărul de telefon indicat este sau nu deja înscris în agenda telefonică.

Date de intrare. Fișierul text TELEFON.IN conține pe prima linie un număr natural n – numărul inițial de numere de telefon din agenda telefonică a lui Ionel. Pe a doua linie este înscris numărul de telefon, pe care Ionel dorește să îl adauge în agenda telefonului său. Următoarele n rânduri conțin numere de telefon în conformitate cu unul dintre cele trei formate din condiție, care sunt deja în agenda telefonului lui Ionel.

Date de ieșire. Fișierul text TELEFON.OUT pe prima linie va conține numerele de ordine ale numerelor de telefon ce coincid cu numărul de telefon, pe care Ionel dorește să îl adauge în agendă sau NO, dacă nici unul nu coincide.

Restricții:

- $1 \leq n \leq 100$;
- Numerotarea începe de la 1;
- Timpul de execuție nu va depăși 1 s;
- Programul va folosi cel mult 32 MB de memorie operativă.

Exemplu:

TELEFON.IN	TELEFON.OUT	Explicații
3 8(373)8123456 +7-373-8-12-34-56 8-1-2-3-4-5-6 8-373-812	1 2	+7(916)8123456 și 89168123456 - este unul și același număr.

3. COLIERE

Un colier de n ($n \in N^*$) mărgelile albe, roșii, galbene, verzi și negre se poate construi respectând următoarele reguli:

- nu plasăm două mărgelile de aceeași culoare în poziții consecutive;
- nu plasăm mărgelile albe lângă mărgelile galbene și nici mărgelile verzi lângă mărgelile negre;
- nu utilizăm mai mult de $n/2$ mărgelile roșii.

Sarcină. Elaborați un program care să determine numărul de posibilități de alcătuire a colierelor din n mărgelile, care satisfac condițiilor indicate.

Date de intrare. Fișierul text COLIER.IN conține pe prima linie un număr natural n – care reprezintă numărul de mărgelile.

Date de ieșire. Fișierul text COLIER.OUT pe prima linie va conține un număr natural – numărul de posibilități de alcătuire a colierelor.

Restricții:

- $2 \leq n \leq 18$;
- Timpul de execuție nu va depăși 10 s;
- Programul va folosi cel mult 32 MB de memorie operativă.

Exemplu:

COLIER.IN	COLIER.OUT	Explicații // Объяснения
4	128	
2	16	Variantele posibile // Возможные варианты: ra, rg, rn, rv, ar, an, av, gr, gn, gv, nr, na, ng, vr, va, vg

4. CONVERSAȚIA ȘPIONILOR

Doi șpioni comunică între ei cu ajutorul unei programe de codare a textului. Ambii cunosc cheia de codare, care reprezintă o succesiune de cifre zecimale (de la 0 la 9). Codarea decurge în două etape. La prima etapă textul este suprapus cu cheia, astfel încât primului element al textului îi corespunde primul element al cheii; al doilea element al textului – cu al doilea element al cheii, etc. Dacă cheia este mai scurtă decât textul, atunci suprapunerea se ia de la începutul cheii. Codarea presupune **adăugarea** la codul ASCII al caracterului din text valorii numerice a elementului din cheie, dacă poziția elementului din text este **impară** (prima poziție are numărul de ordine 1) și **scăderea** la codul ASCII al caracterului din text a valorii numerice a elementului din cheie pentru cele de pe poziții **pare**. La etapa a doua textul obținut după prima etapă este translat cu n poziții la dreapta. La translatarea cu 1 poziție ultimul element al textului ia poziția primului, primul ia poziția elementului al doilea, ș.a.m.d. Valoarea n este suma cifrelor cheii de codare.

Sarcină. Se cere să se determine mesajul codificat.

Date de intrare. Fișierul de intrare SPION.IN conține pe prima linie cheia de codare, iar pe linia a doua – textul ce trebuie codificat.

Date de ieșire. Fișierul text SPION.OUT va conține pe prima linie mesajul codificat.

Restricții:

- cheia – minim 3 elemente, maxim 5 elemente. Textul: minim 5 elemente, maxim 255 de elemente.
- Timpul de execuție nu va depăși 3 s;
- Programul va folosi cel mult 32 MB de memorie operativă.

Exemplu:

SPION.IN	SPION.OUT	Explicații // Объяснения
241 Imposibil...	hdem,2-Kiqmw	După prima etapă: Kiqmwhdem,2- După a doua etapă (rotim la dreapta cu $2+4+1 = 7$): hdem,2-Kiqmw

Fișierele sursă vor avea denumirea XXXK.PAS, XXXK.C sau XXXK.CPP, unde XXX este codul de înregistrare al concursantului, format din litera U (D) și 2 cifre. K este numărul de ordine al sarcinii. De exemplu U011.pas, U012.pas, U013.pas, U014.pas.

1. УГАДАЙ ЧИСЛО

Влад скрывает номер k ($1 \leq k \leq n$). Компьютер пытается угадать, очевидно, делая как можно меньше проверок. Поскольку число k неизвестно, компьютер всегда выбирает число в середине текущего сегмента (десятичные знаки аннулируются). Если:

- число, выбранное компьютером, больше k , тогда генерируется символ M ;
- число, выбранное компьютером, меньше k , тогда генерируется символ P ;
- число, выбранное комп.-м, равно k , тогда генерируется символ E и алгоритм заканчивается.

Задание. Разработать программу, которая моделирует такую игру.

Входные данные. Текстовый файл GHICI.IN содержит в первой строке 2 натуральных числа n и k – которые представляют собой верхний предел чисел и, соответственно, число, скрываемое Владом.

Выходные данные. Текстовый файл GHICI.OUT в первой строке содержит последовательность символов, генерируемые компьютером.

Ограничения:

- $1 \leq n \leq 1000$; $1 \leq k \leq n$;
- Время выполнения – до 1 с; Максимальный объем памяти – до 32 МБ.

Пример:

GHICI.IN	GHICI.OUT	Объяснения
100 78	PPMME	Компьютер генерирует числа 50, 75, 87, 81, 78. Соответственно, генерируемые символы P ($50 < 78$), P ($75 < 78$), M ($87 > 78$), M ($81 > 78$), E ($78 = 78$).

2. ТЕЛЕФОН

Телефонные номера в адресной книге мобильного телефона имеют один из следующих форматов: $+7<код><номер>$; $8<код><номер>$; $<номер>$, где $<номер>$ – это семь цифр, а $<код>$ – это три цифры или три цифры в круглых скобках. Если код не указан, то считается, что он равен 373. Кроме того, в записи телефонного номера может стоять знак «-» между любыми двумя цифрами. На данный момент в адресной книге телефона Вани записано всего n телефонных номеров, и он хочет записать туда еще один. Но он не может понять, записан ли уже такой номер в телефонной книге или нет. Два телефонных номера совпадают, если у них равны коды и равны номера. Например, $+7(916)0123456$ и 89160123456 - это один и тот же номер.

Задание. Разработать программу определения, если записан или нет уже такой номер в тел. книге.

Входные данные. Текстовый файл TELEFON.IN содержит в первой строке натуральное число n – исходное количество записанных телефонных номеров в адресной книге. Во второй строке входных данных записан номер телефона, который Ваня хочет добавить в адресную книгу своего телефона. В следующих n строках записаны номера телефонов согласно одному из трех приведенных в условии форматов, которые уже находятся в адресной книге телефона Вани.

Выходные данные. Текстовый файл TELEFON.OUT в первой строке содержит порядковые номера телефонных номеров, которые совпадают с тем телефонным номером, который Ваня хочет добавить или NO, если ни один не совпадает.

Ограничения:

- $1 \leq n \leq 100$;
- Перечисление телефонных номеров с начинается с 1;
- Время выполнения – до 1 с; Максимальный объем памяти – до 32 МБ.

Пример:

TELEFON.IN	TELEFON.OUT	Объяснения
3 8(373)8123456 +7-373-8-12-34-56 8-1-2-3-4-5-6 8-373-812	1 2	+7(916)8123456 и 89168123456- один и тот же номер.

Ожерелье из n ($n \in N^*$), белых, красных, желтых, черных и белых бусинок могут быть построены в соответствии со следующими правилами:

- Не размещайте две бусинки одного цвета в последовательных положениях;
- Не размещайте белые бусинки рядом с желтыми бусинками или зеленые бусинки рядом с черными бусинками;
- нельзя использовать больше, чем $n/2$ красных шариков.

Задание. Разработать программу, которая определяет количество возможностей изготовления ожерелий из n бусинок, которые удовлетворяют указанным условиям.

Входные данные. Текстовый файл COLIER.IN содержит в первой строке натуральное число n – представляющее количество бусинок.

Выходные данные. Текстовый файл COLIER.OUT содержит в первой строке натуральное число – количество возможностей изготовления ожерелий.

Ограничения:

- $2 \leq n \leq 18$;
- Время выполнения – до 10 с. Максимальный объем памяти – до 32 МБ.

Пример:

COLIER.IN	COLIER.OUT	Объяснения
4	128	
2	16	Variantele posibile // Возможные варианты: ra, rg, rn, rv, ar, an, av, gr, gn, gv, nr, na, ng, vr, va, vg

4. ДИАЛОГ ШПИОНОВ

Два шпиона общаются друг с другом с помощью программы кодирования текста. Оба знают ключ кодирования, который является последовательностью десятичных цифр (от 0 до 9). Кодирование происходит в два этапа. На первом этапе текст накладывается на ключ т.о., чтобы первый элемент текста соответствовал первому элементу ключа; второй элемент текста - со вторым элементом ключа и т. д. Если ключ короче текста, то наложение берется с начала ключа. Кодирование включает в себя **добавление** к коду ASCII-символа текста числового значения элемента в ключе, если позиция символа текста нечетно (первая позиция пронумерована 1) и **убавлением** к коду ASCII-символа текста числового значения элемента в ключе, если позиция символа текста четно. На втором этапе текст, полученный после первого этапа, вращается на n позиций вправо. Вращение на 1 позицию означает перевод последнего элемента текста на первую позицию; первого элемента текста на вторую позицию; и т.д. Значение n является суммой цифр ключа кодирования.

Задание. Требуется определить закодированное сообщение.

Входные данные. Текстовый файл SPION.IN содержит в первой строке ключ кодирования, а на второй строке – текст, подлежащий кодированию.

Выходные данные. Текстовый файл SPION.OUT содержит в перв. строке закодированное сообщение.

Ограничения: Ключ – минимум 3, максимум 5 элементов. Текст: минимум 5 элементов, максимум 255 элементов.

- Время выполнения – до 3 с; Максимальный объем памяти – до 32 МБ.

Пример:

SPION.IN	SPION.OUT	Explicații // Объяснения
241 Imposibil...	hdem,2-Kiqmw	После первого этапа: Kiqmwhdem,2- После второго этапа (вращаем вправо на $2+4+1 = 7$): hdem,2-Kiqmw

Имена исходных файлов будет XXXK.PAS, XXXK.C или XXXK.CPP, где XXX – регистрационный код ученика из буквы U (D) и 2 цифр. K – номер задачи. Например, U011.pas, U012.pas, U013.pas, U014.pas.