Zad. 1. **Szyfrowanie wiadomości** Adam i Janek ustalili między sobą, że każdą wiadomość tekstową, będą szyfrować przy pomocy następującego sposobu. Każdą literę i znak interpunkcyjny należy zamienić na odpowiadającą liczbę kodu ASCII (liczba naturalna od 0 do 127). Pierwsza, tak powstała liczba, jest zamieniana na system dwójkowy. Kolejne liczby, wynikające z kodu ASCII, są zamieniane na system liczbowy, który jest równy powiększonej o dwa reszcie z dzielenia przez osiem poprzedniej liczby. Ilość cyfr zamienionej liczby, dla każdego systemu liczbowego, wynika z ilości cyfr zamiany liczby maksymalnej, czyli liczby 127 (dla systemu binarnego jest to siedem cyfr).

Twoim zadaniem jest napisanie programu, który szyfruje i deszyfruje wiadomości.

Szyfrowanie tekstu z pliku *tekst.txt* do pliku *szyfr.txt*

Deszyfrowanie tekstu z pliku *szyfr.txt* do pliku *odszyfrowane.txt*

Przykład

tekst kod ASCII system liczbowy: 2, (66 mod 8)+2=4, (105 mod 8)+2=5

Bit → 66 105 116 → 1000010 1221 0431

Zad. 2. **Kółko i krzyżyk** Napisz program, który grał będzie z użytkownikiem w klasyczne (dziewięciopolowe) kółko i krzyżyk. Ponieważ grając ,,rozsądnie” nie można w tej grze przegrać – program nie może, niezależnie od tego kto zaczyna i w jaki sposób stawiał będzie swoje znaki, przegrać. Oczywiście, jeśli przeciwnik nie gra ,,rozsądnie”, można w tej grze wygrać – program musi wykorzystywać błędy przeciwnika i przy złym jego posunięciu musi obrać strategię wygrywającą. Komunikację człowiek-komputer i stronę graficzną pozostawiamy w gestii rozwiązującego.

Zad. 3. **Cze, cizie!** W pliku *słownik.txt* znajduje się słownik, w którym słowa (każde w nowej linii) posortowane są alfabetycznie. Napisz program, który wyszukiwał będzie w tym słowniku słów, takich jak *uczci*: wykreślając z niego co drugą literę, spełnione są trzy warunki: 1) skreślana litera jest literą wewnętrzną danego słowa; 2) skreślane litery są jednakowe; 3) pozostały wyraz również jest w słowniku. W przykładzie skreślamy dwa razy literę *c*, a pozostały wyraz – *uzi,* podobnie jak dane słowo *uczci*,znajduje się w tym słowniku. Postaraj się aby Twój algorytm był optymalny (szybkość działania).

Zad. 4 **Biorę np.** Permutacją danego zbioru nazywany dowolne ustawienie w ciąg wszystkich (każdy element wykorzystany jest dokładnie raz) elementów tego zbioru. Nauczyciel matematyki przyjął następujący system odpytywania uczniów na lekcji: pierwsza pytana osoba jest ustalona przez losową liczbą naturalną *n* (np. numerem dnia miesiąca) – jeśli *n* jest niewiększa od liczby osób w klasie *m*, to pytana jest osoba o numerze *n*, w przeciwnym przypadku nauczyciel przyjmuje, że po osobie nr *m* kolejną jest znowu osoba nr 1. Kolejna pytana osoba na numer powiększony o *n* (pamiętaj o ,,zapętlaniu się” listy). Nauczyciel pyta *m* razy. Przykładowo, jeśli w klasie jest 5 osób (*m*=5), a nauczyciel wylosuje liczbę 3 (*n*=3), to do kolejność odpytywania jest następująca: 3, 1, 4, 2, 5. Jeśli w klasie byłoby 6 osób, to kolejność odpytywania (dla tego samego n) byłaby następująca: 3, 6, 3, 6, 3, 6. W pierwszym przypadku kolejność odpytywania jest permutacją numerów osób na liście, a w drugim nie jest (brakuje czterech osób z listy).

Napisz program, który dla zadanych wartości *n* i *m* ustali kolejność pytania uczniów i wypisze ją, jeśli jest ona permutacją.

Zastanów się (i odpowiedz) czy:

1. da się powiedzieć (bez korzystania z programu) czy dla ustalonych wartości liczb naturalnych *n* i *m* otrzymamy permutację?
2. ile można otrzymać (nie interesuje nas wartość liczbowa, tylko opis słowny tej wielkości) różnych permutacji dla ustalonego *m* i różnych wartości *n*?
3. istnieje liczba naturalna *m* taka, że da się dobrać dla niej pewne wartości *n*, takie, że wygenerujemy wszystkie możliwe permutacje zbioru ?

Zad. 5. **Pole boczne bryły obrotowej** Biorąc pewną krzywą dla , która przyjmuje na tym przedziale nieujemne wartości i obracając ją wokół osi *OX*, otrzymamy pewną bryłę obrotową. Pole powierzchni (w tym przypadku bryła jest otwarta z obydwu stron – mamy więc do czynienia z jej powierzchnią boczną) tej bryły można obliczyć następująco: podzielmy przedział na *n* części równoodległymi punktami , (pierwszy z tych punktów ma wartość *a,* ostatni *b*). Oznaczmy przez wartość Łącząc każde dwa kolejne punkty odcinkiem, otrzymamy ramiona (nieprostopadłe do podstaw) trapezów prostokątnych, które obracając się wokół osi *OX* (czyli wokół ramienia prostopadłego do jego podstaw) utworzą stożki ścięte, których pola powierzchni bocznych sumują się do przybliżenia pola powierzchni bocznej wyjściowej bryły. Napisz program, który dla zadanej funkcji , danego przedziału i danej gęstości jego podziału *n* zwracał będzie przybliżone pole powierzchni bocznej otrzymanej bryły obrotowej. (Program możesz testować dla znanych Ci brył obrotowych, np. biorąc otrzymamy, po obrocie, sferę jednostkową o polu .)

