Problema de nota 10

Alice si Bob vor sa comunice sigur, astfel ca utilizeaza structuri algebrice pentru crearea unui mod de a cripta mesajele. Modul lor de lucru se bazeaza pe urmatorul argument matematic: daca p este un numar prim, atunci grupul multiplicativ al unitatilor $(\mathbb{Z}_p^*,\cdot)$ este ciclic si este izomorf cu grupul aditiv $(\mathbb{Z}_{p-1},+)$. Izomorfismul este dat imediat daca se cunoaste un generator al grupului multiplicativ: fie $g \in \mathbb{Z}_p^*$ un generator (i.e. $\mathbb{Z}_p^* := < g >$; g este generator daca si numai daca puterile lui $\bmod p$ genereaza toate elementele din grupul multiplicativ al unitatilor), atunci fiecarui element din $\mathbb{Z}_{p-1}$ ii corespunde un element din $\mathbb{Z}_p^*$ dupa regula:

$$\forall x \in \mathbb{Z}_{p-1}\ \exists!\, y = g^{x-1} \in \mathbb{Z}_p^*$$

De exemplu, daca avem $p = 7$, adica grupurile $(\mathbb{Z}_7^*,\cdot)$ si $(\mathbb{Z}_6,+)$, trebuie sa avem o corespondenta biunivoca intre elementele din $\mathbb{Z}_6$ si cele din $\mathbb{Z}_7^*$. Conform definitiei de mai sus, daca stim ca g este un generator pentru $\mathbb{Z}_7^*$, atunci lui 0 din $\mathbb{Z}_6$ ii corespunde $g^0 = 1$ din $\mathbb{Z}_7^*$, lui 1 din $\mathbb{Z}_6$ ii corespunde $g^1 = g$ din $\mathbb{Z}_7^*$, ..., lui 5 din $\mathbb{Z}_6$ ii corespunde $g^5 \bmod p$ din $\mathbb{Z}_7^*$.

O codificare este, in acest caz, translatarea tuturor elementelor din grupul aditiv in elementele corespunzatoare din grupul multiplicativ, conform izomorfismului de mai sus. In loc de mesajul ACAD, de exemplu, care s-ar scrie $(0,2,0,3)$ (conform pozitiilor in alfabet), am obtine mesajul $(g^0 = 1, g^2, g^0 = 1, g^3)$, adica un mesaj de forma B_B_ (necunoscand inca cine este g, nu stim inca mesajul final).

Determinarea lui g este simpla pentru numere prime mici: se cauta, pe rand, toate elementele din grupul multiplicativ al unitatilor care, ridicate succesiv la putere, genereaza toate elementele grupului. Este evident ca nu se va testa si elementul 1, pentru ca nu va produce nicio modificare ridicarea lui la putere.

Sa consideram cazul lui $\mathbb{Z}_7^*$, un generator al lui poate fi oricare dintre elementele $\{2,3,4,5,6\}$. Chiar daca un grup poate avea mai multi generatori, ne intereseaza doar cel mai mic (adica verificam, in ordine, de la 2 la 6, si alegem astfel).

Incercam elementele pana obtinem generatorul. Toate calculele de mai jos sunt facute modulo 7:

Element	Puterea 0	Puterea 1	Puterea 2	Puterea 3	Puterea 4	Puterea 5	Puterea 6
2	1	2	4	1 (8 mod 7)	nu mai are sens verificarea, daca am obtinut deja 1, de aici se vor repeta valorile, deci 2 nu este generator		
3	1	3	2 (9 mod 7)	6	4	5	1

Am obtinut, deci, izomorfismul

Element din $\mathbb{Z}_6$	0	1	2	3	4	5	6
Element din $\mathbb{Z}_7^*$	1	3	2	6	4	5	1

Acum stim ca mesajul (ACAD) = (0,2,0,3) se traduce in (1,2,1,6) = BCBG

Cerintele temei:

1. Se citeste un numar p de la tastatura. Sa se determine generatorul (cel mai mic) g astfel incat $\mathbb{Z}_p^* :=< g >$ (adica sa se determine generatorul cu proprietatea de mai sus: ridicat succesiv la putere, modulo numarul citit p, sa genereze toate elementele de la 1 la $p-1$). Atentie! Nu se garanteaza faptul ca p este dat mereu corect (adica numar prim), trebuie testata initial aceasta conditie. Daca p NU este prim, se va afisa un mesaj corespunzator pe ecran, altfel se va afisa un mesaj de forma (Generatorul g este: ...).
2. Dat un mesaj clar, sa se cripteze.
3. Dat un mesaj criptat, sa se decripteze.

Datele de intrare vor fi date simultan, cate una pe linie (deci in Input vor fi 3 linii: numarul p, mesajul clar, mesajul criptat).

Observatii privind notarea subpunctelor

1. Valoreaza maxim 3 puncte, care se vor obtine daca si numai daca verificarea numarului prim este eficienta (raportata la impartirea pe intregi: nu se vor verifica elementele de la $\left[\frac{p}{2}\right]+1$ cand se vor cauta divizorii) si, pentru a calcula g, se va retine mereu ultima putere $\bmod p$, adica vom calcula astfel:

   $$g^k \bmod p = g \cdot \left(g^{k-1} \bmod p\right) \bmod p = g \cdot g_{pas\_anterior} \bmod p$$

   si **nu** sub forma

   $$g^k \bmod p = g \cdot \ldots \cdot g \bmod p \ (de\ k\ ori)$$

   Pentru incalcarea acestor optimizari se obtin maxim 1.7 puncte, raportate la functionalitatea acestei prime parti.

Atentie si la verificarea faptului ca s-au obtinut toate numerele din grupul multiplicativ al unitatilor! O varianta ar fi sa sortati vectorul (intr-o copie) si sa verificati ca elementele sunt intr-o ordine **strict crescatoare**. Altfel, poate fi cautat, secvential, fiecare element in parte.

2. Pentru subpunctele 2 si 3 se acorda restul de maxim 3 puncte pana la 10. Exista doua moduri in care puteti citi mesajele:
   a. Cel simplu, in care nu dati mesaj text, ci mesaj numeric, adica in loc de mesajul ACAD veti da in zona de input

      4

      0

      2

      0

      3

      (4 pentru lungimea mesajului, 0 corespunzator lui A, 2 lui C, 0 lui A, 3 lui D)
   b. Cel complex, in care veti da direct mesajul ACAD.

   Daca veti opta pentru cazul simplu, puteti obtine maxim 1.3 puncte, raportate la functionalitatea programului. Daca optati pentru cazul complex, trebuie sa utilizati urmatorul apel sistem pentru READ STRING, descris informal: se incarca in $a0 o zona de memorie unde veti stoca sirul de caractere, in $a1 se incarca lungimea maxima a sirului de caractere citit, se da in $v0 codul 8 si se apeleaza sistemul. Dupa apelul sistem, sirul de caractere este stocat la nivel de memorie in zona alocata si incarcata ca adresa in registrul $a0.

Alte observatii:

1. Evitati sa utilizati registri suplimentari acolo unde nu este cazul. Se depuncteaza pana in 0.5 puncte utilizarea, per total, a unor registri suplimentari care puteau fi evitati.
2. Evitati sa parcurgeti inutil structurile de date (vectorii pe care ii retineti si sirurile de caractere). Daca puteti rezolva dintr-o parcurgere mai multe operatii, procedati astfel. Se depuncteaza pana in 0.5 puncte parcurgerile inutile.
3. NU este necesara utilizarea procedurilor pentru aceasta tema! Tot programul poate fi scris in main fara sa existe depunctari!

Indicatii: pentru a fixa un alfabet de lucru, puteti declara, in zona de data, un sir de forma:

alfabet: .asciiz "ABCDE..."

unde treceti literele cu care vreti sa lucrati, eventual, pentru testarea programului. In acest caz, daca $t0 trece in $t1 prin izomorfism, inseamna ca alfebet($t0) trece in alfabet($t1). (retinerea unui sir de caractere *alfabet* va ajuta sa scapati de lucrul pe ASCII-uri, unde trebuia sa adunati un 64 si, in plus, va permite sa puneti literele pe care le vreti voi – de exemplu, daca vrem numarul 23, prim, putem testa mesaje din alfabet renuntand la 3 litere – in special q, w, y, care nu sunt foarte frecvente in limba romana, de exemplu).

Atentie ca ordinul unui element divide ordinul grupului, puteti folosi aceasta informatie pentru a determina si mai rapid un generator.

Doar citirea si afisarea unor date neprelucrate nu sunt operatii suficiente pentru a obtine 5 pe aceasta tema!