Tema 1 laborator ASC

Descrierea soluției:

Pentru citirea datelor de intrare, am folosit procedurile **citire_numar**, **citire_mesaj_clar** si **citire_mesaj_criptat**.

Urmează să verificăm dacă numărul **p** din cerință este prim cu ajutorul procedurii **verif_prim**. În procedura, ne folosim de registrul **$t0** pentru a parcurge toți potențialii divizori de la 2 până la $\left[\frac{p}{2}\right]$ și verificăm dacă am găsit un divizor prin efectuarea restului împărțirii lui **p** la **$t0**. Dacă restul este 0, atunci am găsit un divizor al lui p și încărcăm valoarea 1 în registrul **$t2**. În final, dacă registrul **$t2** are valoarea 0, înseamnă că nu am găsit niciun divisor al lui **p**, deci numărul este prim și procedura întoarce valoarea 1, iar în caz contrar, întoarce valoarea 0.

Dacă procedura a întors valoarea 0, atunci apelăm procedura **exit_nonprim** care afișează un mesaj și încheie execuția programului. În schimb, dacă numărul este prim, este garantat faptul că grupul $\left(\mathbb{Z}_p^*, \cdot\right)$ este ciclic, deci poate fi generat de un singur element.

Astfel, apelăm procedura **cauta_generator**, care determină și afișează valoarea generatorului, și, de asemenea, va stoca în vectorul **v**, reținut la nivel de memorie, valorile generate în ordine de către generator. În procedură, ne folosim de registrul **$s1** pentru a parcurge fiecare număr **g** de la 2 la p-1 și a genera toate puterile până când dăm de elementul neutru(1). După generarea puterilor, verificăm mai întâi dacă ordinul lui g, adică numărul puterilor generate până la elementul neutru, este mai mic decât ordinul grupului(p-1), caz în care incrementăm **g** cu 1 și efectuăm din nou

generarea puterilor pentru noul candidat. Dacă ord(g)=p-1, atunci este posibil că g să fie generator, însă trebuie să ne asigurăm că au fost generate toate elementele grupului. Pentru a verifica acest lucru, apelăm procedura **verif_generator** care va cauta secvențial fiecare element x din intervalul [2,p-1] în vectorul **v**. Marcăm apariția lui x în vectorul **v** cu ajutorul registrului **$t3**, iar în cazul în care după fiecare dintre căutări valoarea regsitrului **$t3** este 1, înseamnă că în vectorul **v** au fost generate, întradevăr, toate elementele grupului, deci afișăm **g** și ne întoarcem în programul principal.

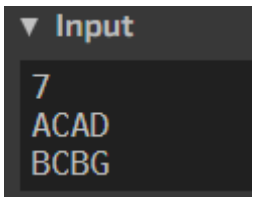
Urmează acum să criptam mesajul clar citit, așa că apelăm procedura **criptare_mesaj_clar**. Acum, pentru a cripta întreg mesajul, vom cripta fiecare litera, pe rând, în felul următor: În registrul **$t3** reținem numărul **i** corespunzător literei, scăzând din codul caracterului codul ASCII al primei litere din alfabet, adică 65. Astfel, pentru litera A avem numărul corespunzător 0, pentru B->1, C->2 etc. Conform izomorfismului, îi asociem numărului **i** puterea $\boldsymbol{g^i}$, unde g este generatorul, iar $\boldsymbol{g^i}$ este chiar elementul de pe poziția **i** din vector(în registrul $t4 reținem **4*i**, adică locația din memorie relativă la **v** a poziției **i** din vector, iar în registrul $t5 încărcăm v[i]). Pentru a obține litera corespunzătoare numărului, adică litera criptată, adunăm cu 65 și stocam caracterul la adresa mesaj_clar_criptat din memorie. Urmând procedeul pentru fiecare litera, în final obținem întreg mesajul criptat la adresa mesaj_clar_criptat, mesaj pe care îl afișăm și apoi ne întoarcem în programul principal.

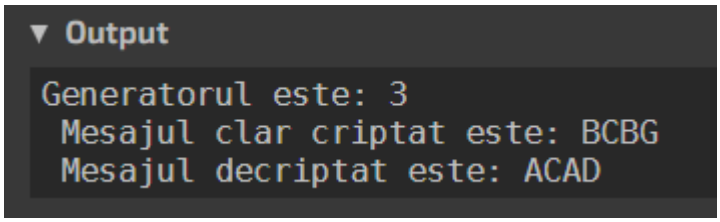
Pentru a decripta al doilea mesaj introdus de la tastatură, vom apela procedura **decriptare_mesaj**, care urmează un procedeu asemănator celui pentru criptare, numai că de data aceasta numărul corespunzător unei litere din mesajul sursă este g^i, iar numărul asociat acestuia este exponentul **i**.

Pentru a determina exponentul **i**, parcurgem vectorul **v** pentru a afla poziția la care se găsește g^i, după care, pentru a obține litera decriptată corespunzătoare, adunăm cu 65 și stocăm caracterul la adresa mesaj_decriptat din memorie. După decriptarea întregului mesaj, îl afișăm și ne întoarcem în programul principal, unde urmează să oprim execuția programului.

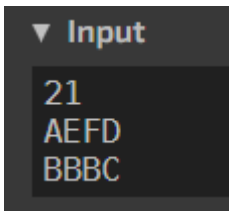
Exemple de date de intrare si de iesire:

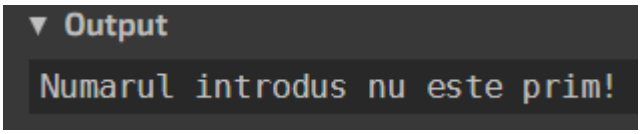
1)





2)





3)