Kodiranje (kodiranje)

Dok ste učili razne enkripcijske algoritme, zainteresovao vas je *One Time Pad* način kodiranja koji je vrlo malo zahtjevan, ali dokazano neprobojan od strane računara (ako je korišten na pravi način). Poruka se šifruje tako što se generiše nasumični ključ iste veličine kao i originalna poruka nakon čega se svaki karakter poruke pretvori u svoju heksadecimalnu *ASCII* predstavu. Na primjer, razmak je $32_{(10)}$ u *ASCII*-u, odnosno $20_{(16)}$ heksadecimalno. Pošto je ključ iste dužine kao i poruka, svaki karakter poruke i ključa se upare, te na njih primijenimo operaciju "ekskluzivnog ili" po bitovima (*XOR*). Ako su bitovi na odgovarajućim mjestima jednaki, onda je rezultat *XOR* operacije nad tim bitovima 0, a ako su različiti onda je 1. Na primjer, ako bi 'c' bio karakter poruke, a '2' odgovarajući karakter u ključu, onda su njihove heksadecimalne vrijednosti redom $63_{(16)}$ i $32_{(16)}$, pa je rezultat *XOR* operacije $51_{(16)}$, jer je 2 *XOR* 3 = 1 i 6 *XOR* 3 = 5 po bitovima. Primijetite da ako se ovaj postupak s istim ključem izvrši nad sada već jednom kodiranim tekstom dobijamo original. Znate da se u poruci mogu nalaziti samo mala slova engleskog alfabeta, te znakovi za tačku '.' ($46_{(10)}$ u *ASCII*) i razmak ' ' ($32_{(10)}$ u *ASCII*), dok se kao ključ mogu koristiti isključivo znakovi koji predstavljaju cifre, tj. od '0' do '9'. Primjer jednog kodiranja bi bio:

abc efg 0120123	61 62 63 20 65 66 67 30 31 32 30 31 32 33	51 53 51 10 54 54 54
Poravnavanje ključa	*ASCII* heksadecimalni	Kodirana poruka

Nakon dosta razmišljanja, dokazali ste da uvijek na osnovu kodirane poruke možete zaključiti na kojim mjestima se tačno nalazio karakter koji je slovo, a na kojima karakter koji je tačka ili razmak, tj. ne možete tačno raspoznati da li je tačka ili razmak, ali ste sigurni da nije slovo.

Zadatak

Vaš zadatak je da napravite program koji implementira sljedeću funkciju:

```
void nadji_slova(int N, int A[], int B[]);
```

N je broj karaktera u poruci za kodiranje, *A* je niz dužine *N* koji predstavlja kodiranu poruku tako što mu je vrijednost *i*-tog člana jednaka *ASCII* vrijednosti karaktera koji se dobije kodiranjem *i*-tog karaktera poruke , dok je *B* niz u koji ćete na *i*-to mjesto upisati broj 1 ako je *i*-ti karakter kodirane poruke bio slovo prije kodiranja, a 0 ako je on bio tačka ili razmak.

Za gore dati primjer, poziv funkcije bi bio:

```
int A[7]={0x51, 0x53, 0x51, 0x10, 0x54, 0x54, 0x54};
```

```
int B[7]={};
nadji_slova(7, A, B);
```

Nakon čega vaš program u niz *B* treba da upiše redom vrijednosti:
`B[0]=1; B[1]=1; B[2]=1; B[3]=0; B[4]=1; B[5]=1; B[6]=1;`

Podzadaci i ograničenja

Vremenska i memorijska ograničenja su dostupna na sistemu za ocjenjivanje.