

Liceul Tehnologic Administrativ și de Servicii
„Victor Slăvescu" Ploiești

Proiect la informatică

Problema nr. 17

Echipa nr.2

Nume și prenume elevi

->State Stefan Alexandru

->Stroie Mihai

->Bocan Silviu Marian

->Golan Teodor Marian

->Gheorge Andrei Laurentu

Clasa a IX a A

2024

Argument

Motivați alegerea problemei (a temei)

Am ales aceasta problema deoarece se bazează pe capacitatea ei de a combina aspecte teoretice și practice ale programării într-o manieră accesibilă și provocatoare. Prin rezolvarea acestei probleme, noi ne-am putut dezvolta și rafina abilitățile de bază necesare în cariera noastră profesională. De asemenea, oferă o înțelegere pragmatică a modului în care se pot aborda și soluționa problemele reale cu ajutorul programării.

Enunțul problemei:

Se consideră că într-un oraș există n ($n \leq 100$) abonați telefonici ale căror numere de telefon sunt formate din 6 cifre, prima cifră fiind obligatoriu nenulă. Presupunem că regia RomTelecom a organizat un concurs al cărui premiu va fi oferit acelui abonat al său, al cărui număr de telefon respectă următoarele condiții:

- se află în lista celor n abonați;
- conține doar cifre pare;
- între numerele care respectă primele două condiții va fi ales numărul care este maxim

Descrierea programului :prezentați datele utilizate, explicați funcțiile și modalitatea de rezolvare.După ce descrieti programul puneți și codul programului.

Date utilizate:

„n” = numarul de numere introduse si verificate;

„c” = variabila folosita la memorarea celui mai mare numar;

„v[100]” = vector folosit la determinarea pozitiei unui numar in sir;

„x” = o copie a numarului prelucrat;

„p” = o variabila folosita la determinarea ultimei cifre a unui numar;

„i”, „j”, „h” = numere folosite la scrierea si citirea vectorului v;

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {   int n,x,v[100],p=10,c=0;
5      cin>>n;
6      if(n<=100&& n>0){
7          for(int i=1; i<=n; i++)
8              cout<<"v["<<i<<"]=";cin>>v[i];
9          for(int j=1; j<=n; j++){
10             if(v[j]>=100000&&v[j]<=999999){
11                 x=v[j];
12                 while(x>0){
13                     if((x%p)%2==0){
14                         x=x/10;
15                     }else{
16                         v[j]=0;
17                         break;
18                     }
19                 }
20             }else{
21                 v[j]=0;
22             }
23             for(int h=1; h<=n; h++){
24                 if(v[h]>c)
25                     c=v[h];
26             }
27         }
28     }else{
29         cout<<"NU";
30     }
31     if(c!=0){
32         cout<<c<<"Este castigatorul";
33     }
34     else{
35         cout<<"nu e castigator";
36     }
37     return 0;
38 }
```

Cerințe minime de sistem, unde veți specifica : procesor, memorie ram, sistem de operare si alte programe necesare pentru a rula aplicația noastră.

Pentru a rula aplicația descrisă, care presupune manipularea și verificarea unor liste de numere de telefon într-un limbaj de programare precum JavaScript/C++, cerințele minime de sistem sunt relativ modeste. Iată specificațiile necesare pentru un sistem capabil să ruleze eficient o astfel de aplicație:

1. Procesor Minim: Procesor dual-core de 1 GHz (de exemplu, Intel Core i3 sau echivalent AMD).
Recomandat: Procesor quad-core de 2 GHz sau mai rapid (de exemplu, Intel Core i5 sau echivalent AMD).
2. Memorie RAM Minim: 2 GB RAM.Recomandat: 4 GB RAM sau mai mult.
3. Sistem de OperareWindows: Windows 7 sau o versiune ulterioară.macOS: macOS 10.10 (Yosemite) sau o versiune ulterioară.Linux: O distribuție modernă (de exemplu, Ubuntu 18.04 sau mai recent, Fedora, Debian).
4. Programe Necesare pentru JavaScript: Este necesar pentru a rula JavaScript în afara unui browser. Descărcați și instalați cea mai recentă versiune JavaScript:Visual Studio
5. Spațiu de StocareMinim: 500 MB liber pentru instalarea JavaScript și a editorului de text.

Utilitatea aplicatiei :

Aplicatia prezentata are multiple utilizări practice, atât pentru utilizatori individuali cât și pentru organizații. În esență, aplicația selectează numărul de telefon câștigător dintr-o listă de abonați, conform unor reguli specifice. Aceasta poate fi folosită într-un context de concurs, dar și în alte scenarii care necesită filtrarea și selecția unor date pe baza anumitor criterii.

Prin modificarea criteriilor de selecție, aplicația poate fi ajustată pentru a răspunde nevoilor specifice ale fiecărui domeniu, oferind o soluție automatizată și eficientă pentru filtrarea și selecția datelor.

Cine poate utiliza aplicația și în ce scop:

Companii de telecomunicații Scop: Organizarea de concursuri pentru abonați, promovarea serviciilor și fidelizarea clienților.

Exemplu: RomTelecom sau alte companii de telecomunicații pot utiliza aplicația pentru a identifica numărul de telefon câștigător într-un concurs bazat pe anumite criterii (de exemplu, numărul conține doar cifre pare). Organizatori de concursuri și campanii de marketing Scop: Realizarea de extrageri și selecții automate pe baza unor criterii predefinite. Exemplu: Agențiile de marketing pot folosi aplicația pentru a selecta câștigători în campanii promoționale bazate pe date numerice (de exemplu, numere de înregistrare, coduri de participare).

Modalități ulterioare de dezvoltare:

Aplicația poate fi îmbunătățită și extinsă în mai multe moduri pentru a-i crește funcționalitatea, scalabilitatea și utilitatea. Iată câteva sugestii pentru dezvoltări ulterioare:

- Implementarea unei interfețe grafice prietenoase pentru utilizator, astfel încât aplicația să fie mai accesibilă celor care nu sunt familiarizați cu linia de comandă.
- Filtrare avansată: Adăugarea posibilității de a defini criterii de filtrare multiple și complexe (de exemplu, cifre impare, numere care conțin anumite secvențe etc.). Sortare și grupare: Oferirea opțiunii de a sorta și grupa numerele de telefon după diverse criterii înainte de a aplica filtrul.
- Stocarea datelor într-o bază de date pentru o gestionare mai eficientă a informațiilor despre abonați.
- Automatizare și Programare: Permițând rularea automată a aplicației la intervale de timp specificate (de exemplu, zilnic, săptămânal).
- Performanță: Monitorizarea și îmbunătățirea performanței aplicației prin profiling și optimizări codului.
- Validare: Implementarea unui sistem robust de validare a numerelor de telefon pentru a asigura corectitudinea și integritatea datelor introduse.
- Securitate: Adăugarea de măsuri de securitate pentru a proteja datele sensibile, cum ar fi criptarea datelor stocate și autentificarea utilizatorilor.
- Scalabilitate: Optimizarea algoritmilor pentru a gestiona un număr mai mare de abonați (de exemplu, folosirea tehnicilor de paralelizare).

Bibliografie

Site-uri pentru documentare:

1. JavaScript:

- MDN Web Docs: Unul dintre cele mai complete și bine documentate site-uri pentru JavaScript
- W3Schools
- JavaScript.info: Un ghid modern și cuprinzător pentru învățarea JavaScript-ului, de la bazele limbajului până la concepte avansate.

2. C++:

- cppreference.com: O referință detaliată pentru standardul C++ și biblioteca standard (STL).
- GeeksforGeeks C++: O platformă educațională care oferă articole, exemple de cod și soluții la probleme comune de programare în C++.
- LearnCpp.com: Un ghid detaliat și structurat pentru învățarea C++, de la concepte de bază până la subiecte avansate.

3. Documentare generala si tutoriale:

- StackOverflow: O comunitate de programatori unde puteți găsi răspunsuri la problemele specifice de programare în JavaScript și C++.
- GitHub: O platformă pentru găzduirea codului și colaborarea între dezvoltatori, unde puteți găsi proiecte open-source relevante și exemple de implementare.