

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Вычислительной техники

ОТЧЕТ
по практической работе № 1
по дисциплине «Алгоритмы и Структуры данных»
Тема: «Множества и Классы»

Студент гр. 1308	_____	Мельник Д. А.
Студент гр. 1308	_____	Лепов А. В.
Научный руководитель	_____	Манирагена В.

Санкт-Петербург
2022

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ	3
2. ЗАДАНИЕ (ВАРИАНТ 4)	3
3. ФОРМАЛИЗАЦИЯ ЗАДАНИЯ	3
4. КОНТРОЛЬНЫЕ ТЕСТЫ	3
5. ВРЕМЕННАЯ СЛОЖНОСТЬ (ОЖИДАЕМАЯ И ФАКТИЧЕСКАЯ)	4
6. РЕЗУЛЬТАТЫ ИЗМЕРЕНИЯ ВРЕМЕНИ	4
7. ВЫВОДЫ.....	4
8. СПИСОК ИСТОЧНИКОВ	4
9. ПРИЛОЖЕНИЕ А.....	5
9.1. Работа со списками.....	5
9.2. Работа с массивами.....	10
9.3. Работа с массивом битов.....	13
9.4. Работа с машинными словами.....	15
9.5. Работа с массивом битов. Генерация тестов.....	16
9.6. Работа с машинными словами. Генерация тестов.....	18
9.7. Работа с массивом битов. Временной тест.	19
9.8. Работа с машинными словами. Временной тест.	21

1. ЦЕЛЬ РАБОТЫ

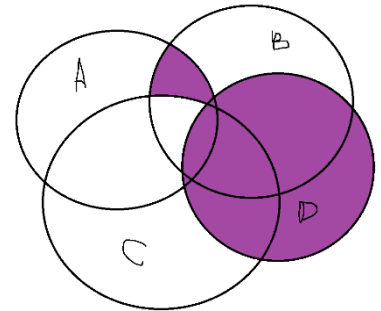
Исследование четырех способов хранения множеств в памяти ЭВМ.

2. ЗАДАНИЕ (ВАРИАНТ 4)

Составить и отладить программу, реализующую обработку множеств.

Строчные латинские буквы.

Множество, содержащее все буквы, общие множества A и B, за исключением букв, содержащих C, а также все буквы D.



3. ФОРМАЛИЗАЦИЯ ЗАДАНИЯ

Формула для вычисления пятого множества по четырем заданным:

$$E = A \cap B \cap \bar{C} \cup D$$

4. КОНТРОЛЬНЫЕ ТЕСТЫ

Исходные данные			
Переменная	Ввод данных	Значение	Тест
A	abcde	abcde	Пройден
B	bdf	bdf	Пройден
C	dfgh	dfgh	Пройден
D	aci	aci	Пройден
Результаты работы программы			
Переменная	Ожидаемый результат	Результат	Тест
E	baci	baci	Пройден

```
"F:\Documents\GitHub\2022-AlgorithmsAndDataStructures\1.1 - Sets\1.1.2-array.exe"
input A : abcde
input B : bdf
input C : dfgh
input D : aci
AandB = bdj
AandBnotC = b!lj
AandBnotCorD = b a c i
Process returned 0 (0x0)   execution time : 53.197 s
Press any key to continue.
```

Рис. 1. Результат работы программы

5. ВРЕМЕННАЯ СЛОЖНОСТЬ (ОЖИДАЕМАЯ И ФАКТИЧЕСКАЯ)

Время выполнения для данной задачи при организации структур данных в виде массивов ожидается наиболее длительным, так как для динамического массива требуется выделение памяти.

Время выполнения при выборе в качестве структуры данных списков ожидается быстрее чем при динамических массивах, но медленнее, чем при массивах битов.

Самой быстрой организацией структур данных для данной задачи ожидаются машинные слова.

6. РЕЗУЛЬТАТЫ ИЗМЕРЕНИЯ ВРЕМЕНИ

Массивы	Списки	Массивы битов	Машинные слова
0.00272с	0.00104сё	0.000115с	0.000014399с

7. ВЫВОДЫ

В результате выполнения лабораторной работы студентами изучены четыре способа хранения множеств в памяти ЭВМ.

8. СПИСОК ИСТОЧНИКОВ

1. Колинко П. Г. // Методические указания по дисциплине «Алгоритмы и структуры данных, часть I» // Санкт-Петербург, СПбГЭТУ «ЛЭТИ», 2022г.
2. Множества в памяти // Алгоритмы и структуры данных. – Лекция.

9. ПРИЛОЖЕНИЕ А

9.1. Работа со списками.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                      //
//    Лепов А. В.                        //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа со списками                //
//                               //
////////////////////////////////////
#include <iostream>
#include <string.h>
#include <algorithm>

using namespace std;

struct Set_node{
    char A;
    Set_node *next;
    Set_node *prev;
};
struct Set_head{
    Set_node *first;
    Set_node *last;
};
Set_head *make_head(){
    Set_head *hed;
    hed = new Set_head;
    hed->first=nullptr;
    hed->last=nullptr;
    return hed;
}
Set_node *make_node(char Arr){
    Set_node *neww;
    neww = new Set_node;

    neww->A = Arr;
    neww->next = nullptr;
    neww->prev = nullptr;
    return neww;
}
void add_node (Set_head *hed, Set_node *neww, Set_node *node0){
    node0->next=neww;
    neww->prev=node0;
    hed->last = neww;
}
void delete_node (Set_node*node1, Set_head *hed){
    Set_node *node0, *node2;
    if ((node1 == hed->first)&&(node1 != hed->last)){
        node2=node1->next;
        hed->first = node2;
    }
}
```

```

    if ((node1 == hed->last)&&(node1 != hed->first)){
        node0=node1->prew;
        hed->last = node0;
    }
    if ((node1 != hed->last)&&(node1 != hed->first)){
        node0=node1->prew;
        node2=node1->next;
        node0->next = node2;
    }

    if ((node1 == hed->last)&&(node1 == hed->first)){
        hed->first = nullptr;
        hed->last = nullptr;
    }
    delete node1;
}

void print_list (Set_head *hed){
    Set_node *node;
    node = hed->first;

    while (node!=hed->last->next){
        cout << node->A << " ";
        node=node->next;
    }
}

void init_list(Set_head *hed, char *C, int n){
    Set_node *neww, *node0;
    int i;
    neww=make_node(C[0]);
    hed->first=neww;
    hed->last=neww;
    node0=hed->last;
    for (i=1;i<n;i++){
        neww = make_node(C[i]);
        add_node(hed,neww,node0);
        node0=neww;
    }
}

void A_and_B(Set_head *W, Set_head *A, Set_head *B){
    Set_node *work1, *work2, *node0, *neww;
    work1 = A->first;
    work2 = B->first;
    neww = make_node('*');
    W->first=neww;
    W->last=neww;
    node0=W->last;
    while (work1!=A->last->next){
        work2 = B->first;
        while (work2!=B->last->next){
            if (work1->A==work2->A){
                neww=make_node(work1->A);
                add_node(W,neww,node0);
                node0=neww;
            }
            work2=work2->next;
        }
    }
}

```

```

        }
        work1=work1->next;
    }
}

void A_and_B_not_C (Set_head *C, Set_head *W){
    Set_node *work1, *work2, *work0;
    int d;
    work1=C->first;
    work2=W->first;

    while (work1!=C->last->next){
        work2=W->first;
        while (work2!=W->last->next){
            d=1;
            if (work1->A == work2->A){
                work0=work2->next;
                delete_node(work2,W);
                work2=work0;
                d=0;
            }
            if (d==1)
                work2=work2->next;
        }
        work1=work1->next;
    }
}

int main(){
    char *notC, *AandB, *AandBnotC, *AandBnotCorD, *Ar, *Cr, *Br, *Dr;
    int i, j, nA, nB, nAB=0, m, nABC=0, nC, nD, nABCD = 0, k=0, d;
    bool kond1, kond2;
    Set_node *neww, *node0, *work1, *work2, *work0, *node1, *node2;;
    Set_head *A, *B, *C, *D, *W;

    A=make_head();
    B=make_head();
    C=make_head();
    D=make_head();

    cout << "input A : ";
    Ar=new char [10];
    cin >> Ar;
    cout << "input B : ";
    Br=new char [10];
    cin >> Br;
    cout << "input C : ";
    Cr=new char [10];
    cin >> Cr;
    cout << "input D : ";
    Dr=new char [10];
    cin >> Dr;

    nA = strlen(Ar);
    nB = strlen(Br);
    nC = strlen(Cr);
    nD = strlen(Dr);

    init_list(A,Ar,nA);

```

```

init_list(B,Br,nB);
init_list(C,Cr,nC);
init_list(D,Dr,nD);

//A and B
W=make_head();
A_and_B(W,A,B);

cout << "AandB = ";
print_list(W);
cout<<endl;

//A anb B without C
A_and_B_not_C(C,W);

cout<<"AandBnotC = ";
print_list(W);
cout << endl;

//A anb B without C or D
work1=W->first;
work2=D->first;

while (work1!=W->last->next){
    work2=D->first;
    while (work2!=D->last->next){
        if (work1->A == work2->A){
            k++;
        }
        work2=work2->next;
    }
    work1=work1->next;
}

work1=W->first;
work2=D->first;

if (k!=0){
    while (work1!=W->last->next){
        work2=D->first;
        while (work2!=D->last->next){
            d=1;
            if (work1->A == work2->A){
                work0=work2->next;
                delete_node(work2,D);
                work2=work0;
                d=0;
            }
            if (d==1)
                work2=work2->next;
        }
        work1=work1->next;
    }
}

node0=W->last;

```



```

work2=D->first;

while (work2!=D->last->next){
    add_node(W,work2,node0);
    node0 = work2;
    work2=work2->next;
}
cout<<"AandBnotCorD = ";
print_list(W);
return 0;
work1=W->first;
work2=W->first;
while (work1!=nullptr){
    work1=work2;
    work2 = work1->next;
    delete work1;
}
delete []Ar;
delete []Br;
delete []Cr;
delete []Dr;
}

```

9.2. Работа с массивами.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                     //
//    Лепов А. В.                       //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с массивами                //
//                               //
////////////////////////////////////
#include <iostream>
#include <string.h>
#include <algorithm>

using namespace std;

int num_of_elements_and (char A[], char B[], int nA, int nB){
    int nAB=0, i, j;
    for (i=0;i<nA;i++){
        for(j=0;j<nB;j++){
            if (A[i]==B[j])
                nAB++;
        }
    }
    return nAB;
}

int main(){
    char A[26], B[26], C[26], D[26], *notC, *AandB, *AandBnotC, *AandBnotCorD;

    int i, j, nA, nB, nAB=0, m, nABC=0, nC, nD, nABCD = 0;

    bool kond1, kond2;

    cout << "input A : ";
    cin >> A;
    cout << "\ninput B : ";
    cin >> B;
    cout << "\ninput C : ";
    cin >> C;
    cout << "\ninput D : ";
    cin >> D;

    nA = strlen(A);
    nB = strlen(B);
    nC = strlen(C);
    nD = strlen(D);

    nAB = num_of_elements_and(A,B,nA,nB);

    //cout << "nAB = " << nAB << endl;

    nAB == 0 ? kond1 = false : kond1 =true;
}
```

```

//cout << "nAB = " << nAB << endl;

if (kond1){
    AandB = new char [nAB];

    m=0;

    for (i=0;i<nA;i++)
    {
        for(j=0;j<nB;j++)
        {
            if (A[i]==B[j])
            {
                AandB[m]=A[i];
                m++;
            }
        }
    }

    cout << "AandB = " << AandB << endl;

    nABC=nAB;

    //cout << "nABC1 = " << nABC << endl;

    for (i=0;i<nAB;i++)
    {
        for (j=0;j<nC;j++)
        {
            if (AandB[i]==C[j])
                nABC--;
        }
    }

    //cout << "nABC2 = " << nABC << endl;

    nABC == 0 ? kond2 = false : kond2 = true;

    if (kond2){

        AandBnotC = new char [nABC];
        for (i=0;i<nAB;i++)
        {
            for (j=0;j<nC;j++)
            {
                if (AandB[i]==C[j])
                    AandB[i]='*';
            }
        }
        m=0;
        for (i=0;i<nAB;i++)
        {
            if (AandB[i]!='*')
            {
                AandBnotC[m]=AandB[i];
                m++;
            }
        }
    }
}

```

```

        }

        cout << "AandBnotC = " << AandBnotC << endl;
    }
} else nABC = nAB;

m=nABC;
nABCD=nABC+nD;

//cout << "nABCD = " << nABCD << endl;

for (i=0;i<nABC;i++){
    for (j=0;j<nD;j++){
        if (AandBnotC[i]==D[j]){
            D[j]='*';
            nABCD--;
        }
    }
}

//cout << "nABCD = " << nABCD << endl;

AandBnotCorD = new char [nABCD];

for (i=0;i<nABC;i++)
{
    AandBnotCorD[i]=AandBnotC[i];
}

for (i=0;i<nD;i++)
{
    if (D[i]!='*'){
        AandBnotCorD[m]=D[i];
        m++;
    }
}

//cout << nABC << endl;
//cout << nABCD << endl;

cout << "AandBnotCorD = ";
for (i=0;i<nABCD;i++)
    cout << AandBnotCorD[i] << " ";

// sort(AandBnotCorD, AandBnotCorD+nABCD);
// for (i=0;i<nABCD;i++)
//     cout << AandBnotCorD[i] << " ";

if (kond1){
    delete []AandB;
    if (kond2)
        delete []AandBnotC;
}
delete []AandBnotCorD;

return 0;
}

```

9.3. Работа с массивом битов.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:  //
//    Мельник Д. А.                    //
//    Лепов А. В.                      //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с массивом битов        //
//                               //
////////////////////////////////////

#include <iostream>
#include<bitset>
#include<iomanip>
#include<string.h>

using namespace std;

bitset<26> into_bitset(char A[]){
    bitset<26> bA=0;
    for (int i=0; i<strlen(A); i++){
        bA[A[i]-'a'] = 1;
    }
    return bA;
}

int main()
{
    bitset<26> bA=0;
    bitset<26> bB=0;
    bitset<26> bC=0;
    bitset<26> bD=0;
    bitset<26> bE=0;
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    char A[10],B[10],C[10],D[10], E[10]="";
    int m = 26;
    cout << "Enter A B C D:" << endl;
    cin >> A >> B >> C >> D;

    bA=into_bitset(A);
    bB=into_bitset(B);
    bC=into_bitset(C);
    bD=into_bitset(D);

    cout << "bA = " << bA << endl;
    cout << "bB = " << bB << endl;
    cout << "bC = " << bC << endl;
    cout << "bD = " << bD << endl;

    //enh

    for (int i=0;i<m;i++){
```

```

        bE[i]=bA[i]*bB[i];
    }
    cout << "A and B = " << bE << endl;

    for (int i=0;i<m;i++){
        bE[i]=bE[i]&&~bC[i];
        if (bE[i]==(-1))
            bE[i]=0;
    }
    cout << "A and B without C = " << bE << endl;

    for (int i=0;i<m;i++){
        bE[i]=bE[i]|bD[i];
    }

    cout << "A and B without C or D = " << bE << endl;

    cout << "result: ";

    for (int i =0, k = 0;i<m;i++){
        if (bE[i]==1){
            E[k]=U[i];
            k++;
        }
    }
    cout << E << endl;
    return 0;
}

```

9.4. Работа с машинными словами.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                      //
//    Лепов А. В.                        //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с машинными словами    //
//                               //
////////////////////////////////////

#include <iostream>

using namespace std;

unsigned int into_word (char A[]){
    unsigned int wA=0;
    for (int i=0; A[i]; ++i){
        wA|=(1<<(A[i]-'a'));
    }
    return wA;
}

void bin_print(unsigned int wA){
    int binstr[32];
    for (int i=0; i<32; i++){
        binstr[i]=0;
        binstr[i]=wA%2;
        wA=wA/2;
    }
    for (int i =31; i>-1; i--)
        cout << binstr[i];
    cout << endl;
}

int main(){
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    char A[10], B[10], C[10]="", D[10], E[10]="";
    unsigned int wA=0, wB=0, wC=0, wD=0, wE=0, wW=0, wE1;
    int m=32;
    cout << "Enter A B C D:" << endl;
    cin >> A >> B >> C >> D;
    wA=into_word(A);
    wB=into_word(B);
    wC=into_word(C);
    wD=into_word(D);

    cout << "A"           = ";
    bin_print(wA);
    cout << "B"           = ";
    bin_print(wB);
    cout << "C"           = ";
    bin_print(wC);
    cout << "D"           = ";
```

```

bin_print(wD);
cout << "AandB      = ";
bin_print(wA&wB);
cout << "AandBnotC   = ";
bin_print ((wA&wB)&~wC);
cout << "AandBnotCorD = ";
bin_print (((wA&wB)&~wC)|wD);

wE=((wA&wB)&~wC)|wD;

for (int i=0, k=0; i<m; i++){
    if ((wE>>i)&1==1){
        E[k++]=U[i];
    }
}
cout << "result = " << E << endl;
}

```

9.5. Работа с массивом битов. Генерация тестов.

```

////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                     //
//    Лепов А. В.                       //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с массивом битов. Генерация тестов.    //
//                               //
////////////////////////////////////
#include <iostream>
#include<bitset>
#include<iomanip>
#include<string.h>

using namespace std;

bitset<26> random_array (int n){
    bitset<26> bA=0;
    int i;
    for (int m=0; m<n; m++){
        i = rand()%26;
        bA[i]=1;
    }
    return bA;
}

void print_word (bitset<26> bE, char U[]){
    char E[26];
    int m=26;
    for (int i =0, k = 0;i<m;i++){
        if (bE[i]==1){
            E[k]=U[i];
            k++;
        }
    }
}

```



```

    }
    cout << E << endl;
}

int main()
{
    srand(time(0));
    bitset<26> bA=0;
    bitset<26> bB=0;
    bitset<26> bC=0;
    bitset<26> bD=0;
    bitset<26> bE=0;
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    int m = 26;

    bA=random_array(5);
    bB=random_array(8);
    bC=random_array(5);
    bD=random_array(2);

    cout << "bA = ";
    print_word(bA,U);
    cout << "bB = ";
    print_word(bB,U);
    cout << "bC = ";
    print_word(bC,U);
    cout << "bD = ";
    print_word(bD,U);

    for (int i=0;i<m;i++){
        bE[i]=bA[i]*bB[i];
    }
    cout << "A and B = ";
    print_word(bE,U);

    for (int i=0;i<m;i++){
        bE[i]=bE[i]&&~bC[i];
        if (bE[i]==(-1))
            bE[i]=0;
    }
    cout << "A and B without C = ";
    print_word(bE,U);
    for (int i=0;i<m;i++){
        bE[i]=bE[i]|bD[i];
    }
    cout << "A and B without C or D = ";
    print_word(bE,U);
    return 0;
}

```

9.6. Работа с машинными словами. Генерация тестов.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                     //
//    Лепов А. В.                       //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с машинными словами. Генерация тестов.    //
//                               //
////////////////////////////////////

#include <iostream>
#include <time.h>

using namespace std;

void print_word(unsigned int wA, char U[]){
    char A[26]="";
    int m=32;
    for (int i=0, k=0; i<m; i++){
        if ((wA>>i)&1==1){
            A[k++]=U[i];
        }
    }
    cout << A << endl;
}

int main(){
    srand(time(0));
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    char E[26]="", A[26]="", B[26]="", C[26]="", D[26]="";
    unsigned int wA=0, wB=0, wC=0, wD=0, wE=0;
    int m=32;

    wA=(long)(rand()%16)|rand();
    wB=(long)(rand()%16)|rand();
    wC=(long)(rand()%16)|rand();
    wD=(long)(rand()%16)|rand();

    cout << "A = ";
    print_word(wA, U);
    cout << "B = ";
    print_word(wB, U);
    cout << "C = ";
    print_word(wC, U);
    cout << "D = ";
    print_word(wD, U);

    wE=((wA&wB)&~wC)|wD; // so small...
    print_word(wE,U);
}
```

9.7. Работа с массивом битов. Временной тест.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:  //
//    Мельник Д. А.                    //
//    Лепов А. В.                      //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с массивом битов. Временной тест.  //
//                               //
////////////////////////////////////

#include <iostream>
#include<bitset>
#include<iomanip>
#include<string.h>

using namespace std;

bitset<26> random_array (int n){
    bitset<26> bA=0;
    int i;
    for (int m=0; m<n; m++){
        i = rand()%26;
        bA[i]=1;
    }
    return bA;
}

void print_word (bitset<26> bE, char U[]){
    char E[26];
    int m=26;
    for (int i =0, k = 0;i<m;i++){
        if (bE[i]==1){
            E[k]=U[i];
            k++;
        }
    }
    cout << E << endl;
}

int main()
{
    bitset<26> bA=0;
    bitset<26> bB=0;
    bitset<26> bC=0;
    bitset<26> bD=0;
    bitset<26> bE=0;
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    int m = 26;

    clock_t t;
    float s=0;

    for (int k=0;k<1000000;k++){
```

```

    t=0;
    t=clock();

    bA=random_array(5);
    bB=random_array(8);
    bC=random_array(5);
    bD=random_array(2);

    // cout << "bA = ";
    // print_word(bA,U);
    //cout << "bB = ";
    //print_word(bB,U);
    //cout << "bC = ";
    //print_word(bC,U);
    //cout << "bD = ";
    //print_word(bD,U);

    for (int i=0;i<m;i++){
        bE[i]=bA[i]*bB[i];
    }
    //cout << "A and B = ";
    //print_word(bE,U);

    for (int i=0;i<m;i++){
        bE[i]=bE[i]&&~bC[i];
        if (bE[i]==(-1))
            bE[i]=0;
    }
    //cout << "A and B without C = ";
    //print_word(bE,U);
    for (int i=0;i<m;i++){
        bE[i]=bE[i]|bD[i];
    }
    //cout << "A and B without C or D = ";
    //print_word(bE,U);
    t=clock()-t;
    s+=t;
    cout<<endl;
}
cout<<"=="<<(s/1000000)*0.017 << endl;
return 0;
}

```

9.8. Работа с машинными словами. Временной тест.

```
////////////////////////////////////
//                               //
//    Выполнили студенты, гр. 1308:    //
//    Мельник Д. А.                     //
//    Лепов А. В.                       //
//                               //
////////////////////////////////////
////////////////////////////////////
//                               //
//    Работа с машинными словами. Временной тест.    //
//                               //
////////////////////////////////////

#include <iostream>
#include <time.h>

using namespace std;

void print_word(unsigned int wA, char U[]){
    char A[26]="";
    int m=32;
    for (int i=0, k=0; i<m; i++){
        if ((wA>>i)&1==1){
            A[k++]=U[i];
        }
    }
    cout << A << endl;
}

int main(){
    // srand(time(0));
    char U[] = "abcdefghijklmnopqrstuvwxyz";
    char E[26]="", A[26]="", B[26]="", C[26]="", D[26]="";
    unsigned int wA=0, wB=0, wC=0, wD=0, wE=0;
    int m=32;
    float s = 0;
    clock_t t;

    for (int k=0; k<1000000; k++){

        t=0;
        t=clock();

        wA=(long)(rand()%16)|rand();
        wB=(long)(rand()%16)|rand();
        wC=(long)(rand()%16)|rand();
        wD=(long)(rand()%16)|rand();

        // cout << "A = ";
        // print_word(wA, U);
        // cout << "B = ";
        // print_word(wB, U);
        // cout << "C = ";
        // print_word(wC, U);
        // cout << "D = ";
        // print_word(wD, U);
    }
}
```

```

        wE=((wA&wB)&~wC)|wD; // so small...
        //print_word(wE,U);
        t=clock()-t;
        s+=t;
        cout<<endl;
    }
    cout << "==" << (s/1000000)*0.017 << endl;
    return 0;
}

```