LAPORAN TUGAS KECIL 1
IF2211 STRATEGI ALGORITMA

Penyelesaian *Cryptarithmetic* dengan Algoritma *Brute Force*

Nama : **Christopher Justine William**
NIM : **13519006**
Kelas : **K-01**
Bahasa : **C**

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2021

A. Algoritma *Brute Force*

Langkah-langkah :

1. Daftarkan setiap huruf yang unik ke dalam suatu tabel S yang maksimal panjangnya 10 elemen.
2. Buatlah sebuah array A dengan panjang 10 elemen yang diisi dengan nilai [0..9].
3. Nilai setiap huruf pada tabel S berkorespondensi dengan array A pada indeks yang sama.
4. Lakukan permutasi pada array A untuk mendapatkan semua kemungkinan solusi.
5. Jika hanya ada kurang dari 10 huruf yang unik pada tabel S, maka permutasinya adalah P(n,k) dengan n = 10 dan k < 10. Nilai huruf pada tabel S yang berkorespondensi dengan array A hanya sampai indeks ke-i, dengan i < n.
6. Untuk setiap hasil permutasi lakukan pengecekan untuk menentukan apakah nilai yang berkorespondensi antara array A dengan tabel S merupakan solusi yang tepat.
7. Program akan berakhir ketika telah menelusuri semua kemungkinan yang ada.

B. Source Code Program

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

/* Definisi Tipe Data */
typedef struct
{
    char Alp;
    int Val;
    int Pos;
} Char;

typedef struct
{
    Char Tab[10];
    int Neff;
} String;

typedef struct
{
    String Tab[10];
    int Neff;
} List;


void inisialisasiPos(List* L)
/* Menginisialisasi posisi setiap character di list */
{
    int i, j;
    for(i = 0; i < L->Neff; i++){
        for(j = 0; j < L->Tab[i].Neff; j++){
            L->Tab[i].Tab[j].Pos = j;
        }
    }
}

void inisialisasiValue(List* L)
/* Menginisialisasi nilai setiap character di list */
{
    int i, j;
    for(i = 0; i < L->Neff; i++){
        for(j = 0; j < L->Tab[i].Neff; j++){
            L->Tab[i].Tab[j].Val = 0;
        }
    }
}
```

```
void printInput(List L)
/* Mencetak input ke layar */
{
    int i, j;
    for(i = 0; i < L.Neff; i++){
        if((i) == L.Neff-2){
            printf("+");
        }
        else{
            printf(" ");
        }
        for(j = 0; j < L.Tab[i].Neff; j++){
            printf("%c", L.Tab[i].Tab[j].Alp);
        }
        printf("\n");
        if(i == L.Neff-2){
            printf(" ");
            for(j = 0; j < L.Tab[i].Neff; j++){
                printf("-");
            }
            printf("\n");
        }
    }
}

void printOutput(List L)
/* Mencetak output ke layar */
{
    int i, j;
    for(i = 0; i < L.Neff; i++){
        if((i) == L.Neff-2){
            printf("+");
        }
        else{
            printf(" ");
        }
        for(j = 0; j < L.Tab[i].Neff; j++){
            if(L.Tab[i].Tab[j].Alp != ' '){
                printf("%d", L.Tab[i].Tab[j].Val);
            }
            else{
                printf(" ");
            }
        }
        printf("\n");
        if(i == L.Neff-2){
            printf(" ");
            for(j = 0; j < L.Tab[i].Neff; j++){
                printf("-");
            }
            printf("\n");
        }
    }
}
```

```
void isiPos(String* S, int arr[])
/* Mengisi String S dengan nilai hasil permutasi yang disimpan di arr */
{
    int i;
    for(i = 0; i < S->Neff; i++){
        S->Tab[i].Val = arr[i];
    }
}

int isMember(String S, char c)
/* Mengecek apakah suatu char merupakan anggota String S */
{
    int i = 0;
    int member = 0;
    while(i < S.Neff && !member){
        if(S.Tab[i].Alp == c){
            member = 1;
        }
        i++;
    }
    return member;
}

void charList(List L, String* S)
/* Mengisi String S dengan char dari List L */
{
    int i, j;
    for(i = 0; i < L.Neff; i++){
        for(j = 0; j < L.Tab[i].Neff; j++){
            if(!isMember(*S, L.Tab[i].Tab[j].Alp) && (L.Tab[i].Tab[j].Alp
!= ' ')){
                S->Tab[S->Neff].Alp = L.Tab[i].Tab[j].Alp;
                S->Neff++;
            }
        }
    }
}

int search(String S, char c)
/* Mereturn nilai dari char c didalam String S */
{
    int i = 0;
    while(i < S.Neff && S.Tab[i].Alp != c){
        i++;
    }
    return S.Tab[i].Val;
}
```

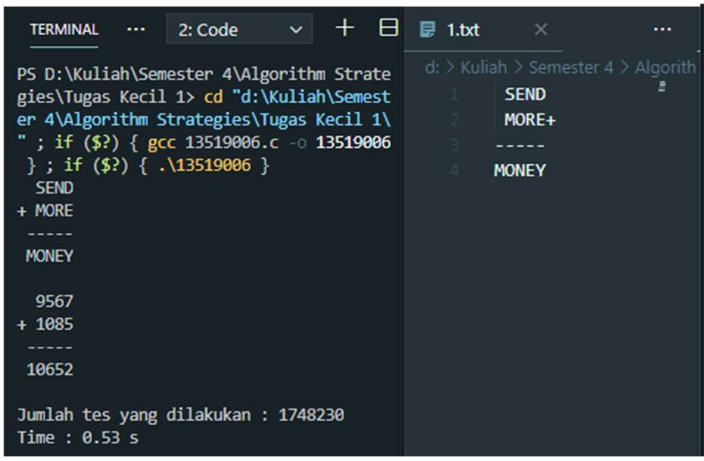
```
void check(List* L, String* S, int* found)
/* Mengecek solusi dari sebuah hasil permutasi */
{
    int i, j;
    for(i = 0; i < L->Neff; i++){
        for(j = 0; j < L->Tab[i].Neff; j++){
            if(L->Tab[i].Tab[j].Alp != ' '){
                L->Tab[i].Tab[j].Val = search(*S,L>Tab[i].Tab[j].Alp);
            }
        }
    }
    if(L->Tab[L->Neff-1].Tab[0].Val == 0){
        return;
    }
    else{
        i = 0;
        int bool = 1;
        while((i < L->Neff-1)&&(bool)){
            j = 1;
            while((j < L->Tab[i].Neff)&&(bool)){
                if((L->Tab[i].Tab[j].Alp != ' ')&&(L->Tab[i].Tab[j-
1].Alp == ' ')&&(L->Tab[i].Tab[j].Alp == 0)){
                    return;
                }
                else if(L->Tab[i].Tab[j].Alp == ' '){
                    j++;
                }
                else{
                    bool = 0;
                }
            }
            i++;
        }
    }
    int val = 0;
    for(j = L->Tab[0].Neff-1; j >= 0; j--){
        for(i = 0; i < L->Neff-1; i++){
            val += L->Tab[i].Tab[j].Val;
        }
        if(val == L->Tab[L->Neff-1].Tab[j].Val){
            val = 0;
        }
        else if(((val%10) == L->Tab[L->Neff-1].Tab[j].Val)&&(j != 0)){
            val = val/10;
        }
        else{
            return;
        }
    }
    *found = 1;
}
```

```
void swap(int arr[], int i, int j)
/* Menukarkan element */
{
    int tmp = arr[j];
    arr[j] = arr[i];
    arr[i] = tmp;
}

void reverse(int arr[], int i, int j)
/* Membalikan urutan element dalam suatu rentang nilai di dalam array*/
{
    swap(arr,i,j);
    if((j-i) > 2){
        reverse(arr,i+1,j-1);
    }
}

void permutasi(int arr[], int n, int k, int* status)
/* Mencari permutasi nilai yang disimpan di dalam array*/
{
    int i = k;
    int limit = k-1;
    while((arr[i] <= arr[limit])&&(i < n)){
        i++;
    }
    if(i < n){
        swap(arr, limit, i);
    }
    else{
        reverse(arr, k, n-1);
        int j = limit-1;
        while((arr[j] >= arr[j+1])&&(j >= 0)){
            j--;
        }
        if(j >= 0){
            i = n - 1;
            while((arr[j] >= arr[i])&&(i > j)){
                i--;
            }
            swap(arr, j, i);
            reverse(arr, j+1, n-1);
        }
        else{
            *status = 0;
            return;
        }
    }
}
```

```
void cryptArithmetic(String* S, List* L, int arr[], int n, int k)
/* Mencari dan menampilkan solusi dari persoalan */
{
    int status = 1, found = 0, count = 1;
    while(status){
        isiPos(S,arr);
        check(L,S,&found);
        if(found){
            printOutput(*L);
            printf("\nJumlah tes yang dilakukan : %d\n", count);
            found = 0;
        }
        if(n == k){
            k--;
        }
        permutasi(arr,n,k,&status);
        count++;
    }
}

int main()
{
    List L;
    L.Neff = 3;
    int i;
    for(i = 0; i < L.Neff; i++){
        L.Tab[i].Neff = 5;
    }

    /* Baca File*/
    char str1[100],str2[100];
    FILE *file = fopen("1.txt", "r");
    if (file == NULL)
    {
        printf("Error!");
        return 1;
    }

    while(fgets(str1, sizeof(str1), file)){
        strcat(str2,str1);
    }
    fclose(file);

    /* Memasukkan hasil baca file ke ADT */
    int j, k, count = 0;
```

```
    for(i = 0; i < L.Neff; i++){
        if(i == L.Neff-1){
            count+= L.Tab[i].Neff+2;
        }
        for(j = 0; j < L.Tab[i].Neff; j++){
            count++;
            L.Tab[i].Tab[j].Alp = str2[count];
        }
        count++;
    }

    /* Start Waktu */
    clock_t t;
    t = clock();

    /* Proses */
    printInput(L);
    printf("\n");

    inisialisasiPos(&L);
    inisialisasiValue(&L);
    int arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
    String S;
    S.Neff = 0;
    charList(L,&S);

    cryptArithmetic(&S,&L,arr, 10, S.Neff);

    t = clock() - t;
    double time_taken = ((double)t)/CLOCKS_PER_SEC;

    printf("Time : %.2f s\n\n", time_taken);

    return 0;
}
```

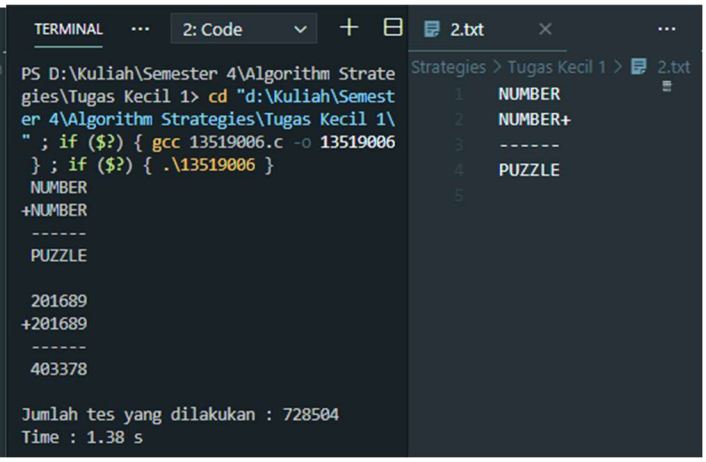
C. *Input* dan *Output*



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
  SEND
+ MORE
 -----
 MONEY

  9567
+ 1085
 -----
 10652

Jumlah tes yang dilakukan : 1748230
Time : 0.53 s
```

1.txt

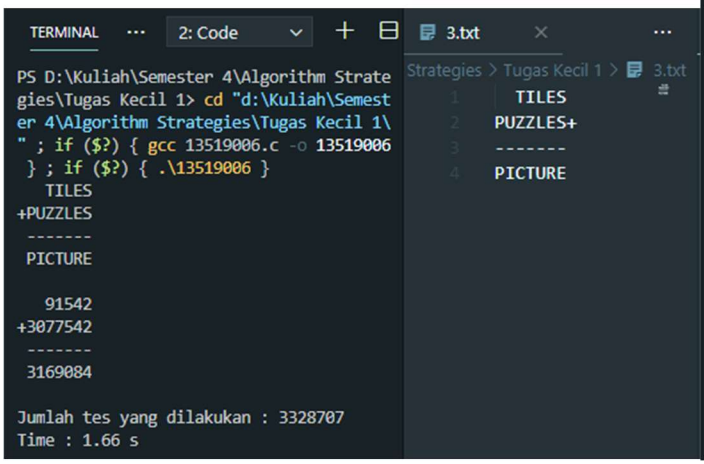
```
SEND
MORE+
-----
MONEY
```



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
 NUMBER
+NUMBER
 ------
 PUZZLE

 201689
+201689
 ------
 403378

Jumlah tes yang dilakukan : 728504
Time : 1.38 s
```

2.txt

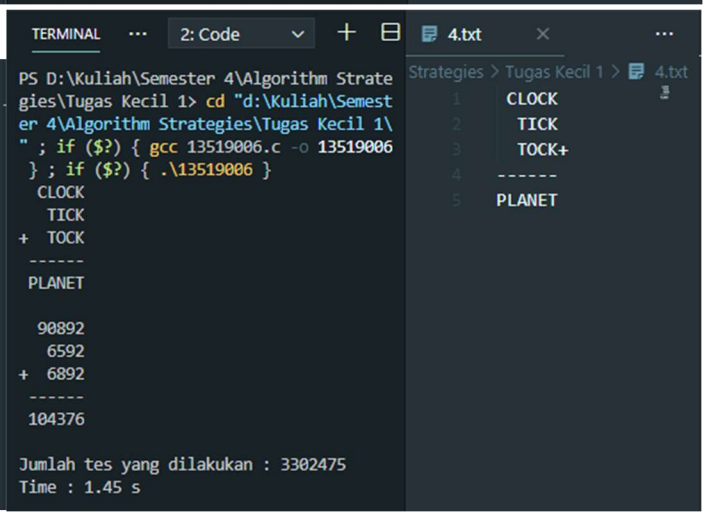
```
NUMBER
NUMBER+
------
PUZZLE
```



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
   TILES
+PUZZLES
 -------
 PICTURE

   91542
+3077542
 -------
 3169084

Jumlah tes yang dilakukan : 3328707
Time : 1.66 s
```

3.txt

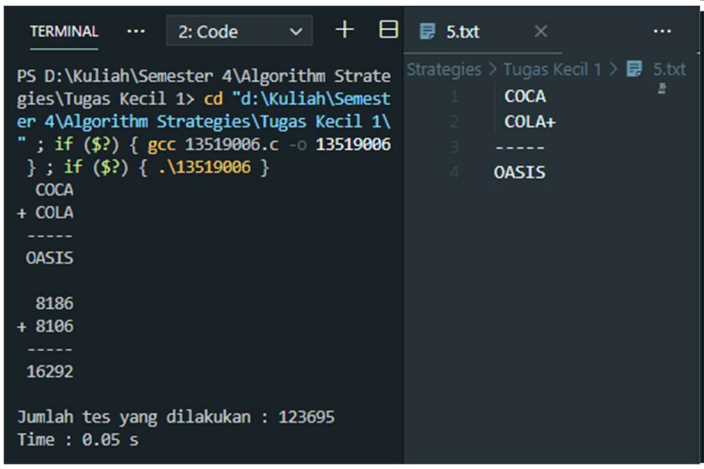
```
TILES
PUZZLES+
-------
PICTURE
```



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
  CLOCK
   TICK
+  TOCK
 ------
 PLANET

  90892
   6592
+  6892
 ------
 104376

Jumlah tes yang dilakukan : 3302475
Time : 1.45 s
```

4.txt

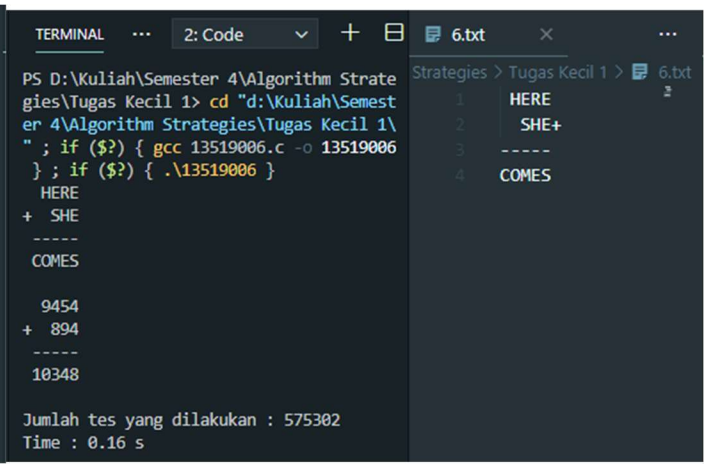
```
CLOCK
TICK
TOCK+
------
PLANET
```



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
  COCA
+ COLA
 -----
 OASIS

  8186
+ 8106
 -----
 16292

Jumlah tes yang dilakukan : 123695
Time : 0.05 s
```

5.txt

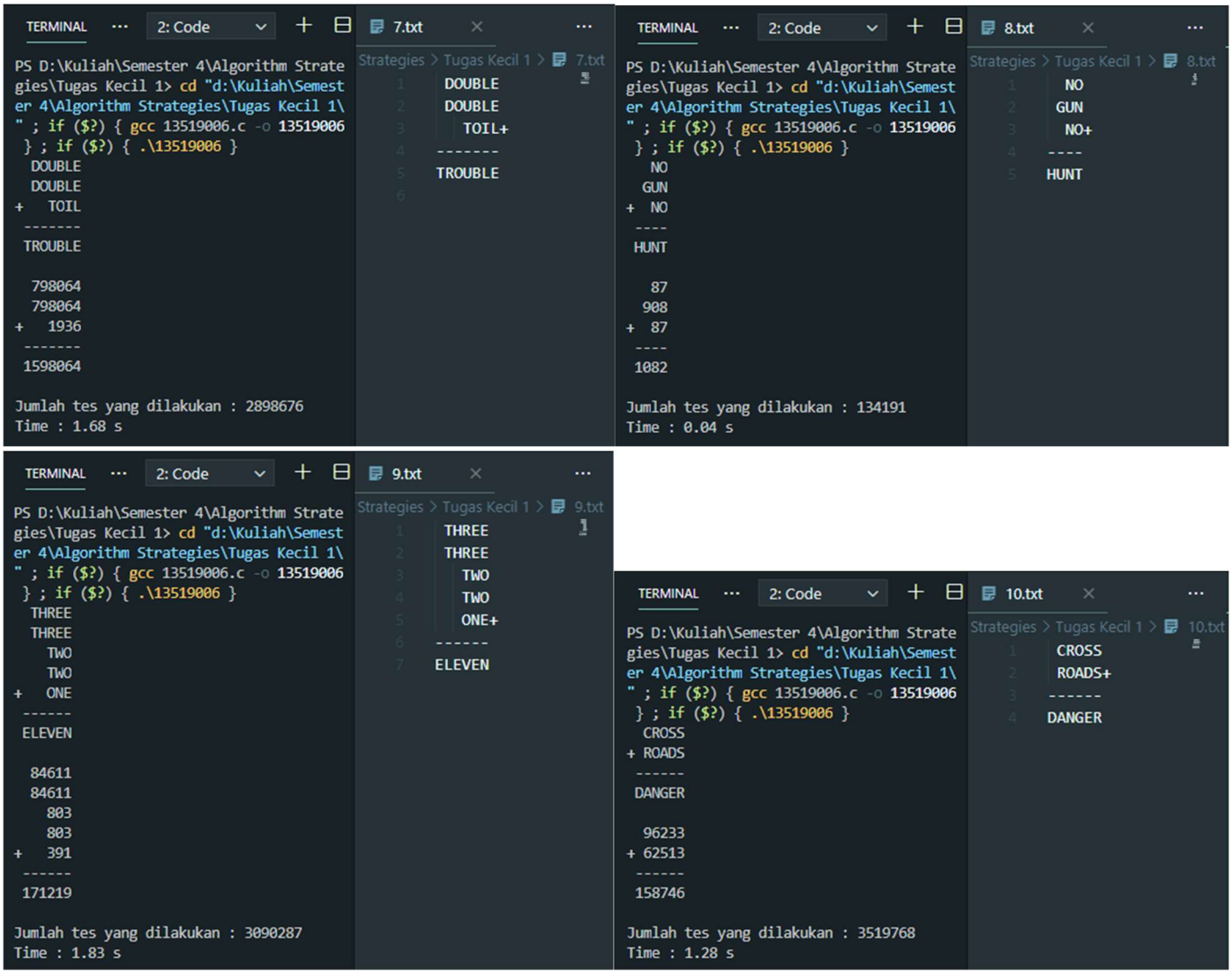
```
COCA
COLA+
-----
OASIS
```



```
PS D:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1> cd "d:\Kuliah\Semester 4\Algorithm Strategies\Tugas Kecil 1\" ; if ($?) { gcc 13519006.c -o 13519006 } ; if ($?) { .\13519006 }
  HERE
+  SHE
 -----
 COMES

  9454
+  894
 -----
 10348

Jumlah tes yang dilakukan : 575302
Time : 0.16 s
```

6.txt

```
HERE
SHE+
-----
COMES
```



D. Drive

https://drive.google.com/drive/folders/1Pn-du_VDrtLbrUVt3l4rDhisULghr1j5?usp=sharing

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan (no syntax error	V	
2. Program berhasil running	V	
3. Program dapat membaca file masukan dan menuliskan luaran.	V	
4. Solusi *cryptarithmetic* hanya benar untuk persoalan *cryptarihtmetic* dengan dua buah *operand*.		V
5. Solusi *cryptarithmetic* benar untuk persoalan *cryptarihtmetic* untuk lebih dari dua buah operand.	V	