

## Домашнє завдання №1

З використанням реалізації поведінки детермінованого кінцевого автомату написати на C/C++ програму(англ. *automata-based programming*) виявлення ділянки тексту у вхідній стрічці. Програма має лише відображати перехід моделі автомату у кінцевий стан без виявлення позиції входження шуканої ділянки тексту.

### Вибір варіанту

Шукана ділянка тексту це прізвище студента записане латинськими літерами.

## Приклад коду

Лістинг 1

```
#include "stdio.h"

#define DECLSTATE(NAME, ...) typedef enum {__VA_ARGS__, size##NAME} NAME;
#define GET_ENUM_SIZE(NAME) size##NAME

DECLSTATE(A,
    vA/*A*/,
    vB/*B*/,
    vC/*C*/,
    vD/*D*/,
    vE/*E*/,
    vF/*F*/,
    vG/*G*/,
    vH/*H*/,
    vI/*I*/,
    vJ/*J*/,
    vK/*K*/,
    vL/*L*/,
    vM/*M*/,
    vN/*N*/,
    vO/*O*/,
    vP/*P*/,
    vQ/*Q*/,
    vR/*R*/,
    vS/*S*/,
    vT/*T*/,
    vU/*U*/,
    vV/*V*/,
    vW/*W*/,
    vX/*X*/,
    vY/*Y*/,
    vZ/*Z*/,
    sZ/*[*/,
    sY/*\*/,
    sC/*]*/,
    sH/*^*/,
    sB/*_*/,
    sT/*`*/,
    va/*a*/,
    vb/*b*/,
    vc/*c*/,
    vd/*d*/,
    ve/*e/>,
```

```

vf/*f*/,
vg/*g*/,
vh/*h*/,
vi/*i*/,
vj/*j*/,
vk/*k*/,
vl/*l*/,
vm/*m*/,
vn/*n*/,
vo/*o*/,
vp/*p*/,
vq/*q*/,
vr/*r*/,
vs/*s*/,
vt/*t*/,
vu/*u*/,
vv/*v*/,
vw/*w*/,
vx/*x*/,
vy/*y*/,
vz/*z*/,
sF/*end mark*/
)

```

```

DECLSTATE(Q,
q0,
q1,
q2,
q3,
q4,
qf
)

```

```

typedef unsigned char INSTRUCTION;
typedef INSTRUCTION PROGRAM[GET_ENUM_SIZE(A)][GET_ENUM_SIZE(Q)];

```

```

PROGRAM program = {
//      q0  q1  q2  q3  q4  qf
/*A*/{q0, q0, q0, q0, q0, qf},
/*B*/{q0, q0, q0, q0, q0, qf},
/*C*/{q0, q0, q0, q0, q0, qf},
/*D*/{q0, q0, q0, q0, q0, qf},
/*E*/{q0, q0, q0, q0, q0, qf},
/*F*/{q0, q0, q0, q0, q0, qf},
/*G*/{q0, q0, q0, q0, q0, qf},
/*H*/{q0, q0, q0, q0, q0, qf},
/*I*/{q0, q0, q0, q0, q0, qf},
/*J*/{q0, q0, q0, q0, q0, qf},
/*K*/{q1, q0, q0, q0, q0, qf},
/*L*/{q0, q0, q0, q0, q0, qf},
/*M*/{q0, q0, q0, q0, q0, qf},
/*N*/{q0, q0, q0, q0, q0, qf},
/*O*/{q0, q0, q0, q0, q0, qf},
/*P*/{q0, q0, q0, q0, q0, qf},
/*Q*/{q0, q0, q0, q0, q0, qf},
/*R*/{q0, q0, q0, q0, q0, qf},
/*S*/{q0, q0, q0, q0, q0, qf},
/*T*/{q0, q0, q0, q0, q0, qf},
/*U*/{q0, q0, q0, q0, q0, qf},
/*V*/{q0, q0, q0, q0, q0, qf},
/*W*/{q0, q0, q0, q0, q0, qf},
/*X*/{q0, q0, q0, q0, q0, qf},
/*Y*/{q0, q0, q0, q0, q0, qf},
/*Z*/{q0, q0, q0, q0, q0, qf},
/* */{q0, q0, q0, q0, q0, qf},

```

```

/*\*/{q0, q0, q0, q0, q0, qf},
/*]/{q0, q0, q0, q0, q0, qf},
/*^*/{q0, q0, q0, q0, q0, qf},
/*_*/{q0, q0, q0, q0, q0, qf},
/*`*/{q0, q0, q0, q0, q0, qf},
/*a*/{q0, q0, q0, q4, q0, qf},
/*b*/{q0, q0, q0, q0, q0, qf},
/*c*/{q0, q0, q0, q0, q0, qf},
/*d*/{q0, q0, q0, q0, q0, qf},
/*e*/{q0, q0, q0, q0, q0, qf},
/*f*/{q0, q0, q0, q0, q0, qf},
/*g*/{q0, q0, q0, q0, q0, qf},
/*h*/{q0, q0, q0, q0, q0, qf},
/*i*/{q0, q0, q0, q0, q0, qf},
/*j*/{q0, q0, q0, q0, q0, qf},
/*k*/{q0, q0, q0, q0, qf, qf},
/*l*/{q0, q0, q0, q0, q0, qf},
/*m*/{q0, q0, q0, q0, q0, qf},
/*n*/{q0, q0, q0, q0, q0, qf},
/*o*/{q0, q2, q0, q0, q0, qf},
/*p*/{q0, q0, q0, q0, q0, qf},
/*q*/{q0, q0, q0, q0, q0, qf},
/*r*/{q0, q0, q0, q0, q0, qf},
/*s*/{q0, q0, q0, q0, q0, qf},
/*t*/{q0, q0, q0, q0, q0, qf},
/*u*/{q0, q0, q0, q0, q0, qf},
/*v*/{q0, q0, q0, q0, q0, qf},
/*w*/{q0, q0, q0, q0, q0, qf},
/*x*/{q0, q0, q0, q0, q0, qf},
/*y*/{q0, q0, q0, q0, q0, qf},
/*z*/{q0, q0, q3, q0, q0, qf}
};

typedef struct structDFA{
    unsigned char * data;
    PROGRAM * program;
    void(*run)(struct structDFA * dfa);
    Q state;
} DFA;

void runner(DFA * dfa){
    for (; *dfa->data != sF; ++dfa->data){
        dfa->state = (*dfa->program)[*dfa->data][dfa->state];
    }
}

#define MAX_TEXT_SIZE 256
int main(){
    unsigned char data[MAX_TEXT_SIZE] = "Kozak Nazar Bohdanovych", *data_ = data;
    DFA dfa = { data, (PROGRAM*)program, runner, q0 };

    for (; *data_ *data_ -= 'A', *data_ %= GET_ENUM_SIZE(A), ++data_);
    *data_ = sF;
    dfa.run(&dfa);

    if (dfa.state == qf){
        printf("DFA: finit state\r\n");
    }
    else{
        printf("DFA: no finit state\r\n");
    }

    getchar();
    return 0;
}

```

## Лістинг 2

```
#include "stdio.h"

#define DECLSTATE(NAME, ...) typedef enum {__VA_ARGS__, size##NAME} NAME;
#define GET_ENUM_SIZE(NAME) size##NAME

DECLSTATE(A,
    vA/*A*/,
    vB/*B*/,
    vC/*C*/,
    vD/*D*/,
    vE/*E*/,
    vF/*F*/,
    vG/*G*/,
    vH/*H*/,
    vI/*I*/,
    vJ/*J*/,
    vK/*K*/,
    vL/*L*/,
    vM/*M*/,
    vN/*N*/,
    vO/*O*/,
    vP/*P*/,
    vQ/*Q*/,
    vR/*R*/,
    vS/*S*/,
    vT/*T*/,
    vU/*U*/,
    vV/*V*/,
    vW/*W*/,
    vX/*X*/,
    vY/*Y*/,
    vZ/*Z*/,
    sZ/*[*/,
    sY/*\*/,
    sC/*]*/,
    sH/*^*/,
    sB/*_*/,
    sT/*`*/,
    va/*a*/,
    vb/*b*/,
    vc/*c*/,
    vd/*d*/,
    ve/*e*/,
    vf/*f*/,
    vg/*g*/,
    vh/*h*/,
    vi/*i*/,
    vj/*j*/,
    vk/*k*/,
    vl/*l*/,
    vm/*m*/,
    vn/*n*/,
    vo/*o*/,
    vp/*p*/,
    vq/*q*/,
    vr/*r*/,
    vs/*s*/,
    vt/*t*/,
    vu/*u*/,
    vv/*v*/,
```

```

vw/*w*/,
vx/*x*/,
vy/*y*/,
vz/*z*/,
sF/*end mark*/
)

DECLSTATE(Q,
q0,
q1,
q2,
q3,
q4,
qf
)

typedef unsigned char INSTRUCTION;
#define MAX_RULE_COUNT 16
typedef struct {
    unsigned char input;
    INSTRUCTION toState;
} SimpleRule;
typedef SimpleRule PROGRAM[GET_ENUM_SIZE(Q)][MAX_RULE_COUNT];

#define DEFAULT GET_ENUM_SIZE(A)

PROGRAM program = {
    //          q0                      q1                      q2
q3          { { vk, q1 }, { DEFAULT, q0 } }, { { vo, q2 }, { DEFAULT, q0 } }, { { vz, q3 }, {
DEFAULT, q0 } }, { { va, q4 }, { DEFAULT, q0 } }, { { vk, qf }, { DEFAULT, q0 } }, { {
DEFAULT, qf } },
};

typedef struct structDFA{
    unsigned char * data;
    PROGRAM * program;
    void(*run)(struct structDFA * dfa);
    Q state;
} DFA;

void runner(DFA * dfa){
    for (; *dfa->data != sF; ++dfa->data){
        SimpleRule *prevRule, *rule;
        for (prevRule = rule = (*dfa->program)[dfa->state]; prevRule->input != DE-
FAULT; prevRule = rule, ++rule){
            if (DEFAULT == rule->input){
                dfa->state = rule->toState;
                break;
            }
            else if (*dfa->data == rule->input){
                dfa->state = rule->toState;
                break;
            }
        }
    }
}

#define MAX_TEXT_SIZE 256
int main(){
    unsigned char data[MAX_TEXT_SIZE] = "Kozak Nazar Bohdanovych", *data_ = data;
    DFA dfa = { data, (PROGRAM*)program, runner, q0 };

    for (; *data_; *data_ -= 'A', *data_ %= GET_ENUM_SIZE(A), ++data_);
    *data_ = sF;
}

```

```
dfa.run(&dfa);

if (dfa.state == qf){
    printf("DFA: finit state\r\n");
}
else{
    printf("DFA: no finit state\r\n");
}

getchar();
return 0;
}
```