

СИСТЕМА ПОСЛІДОВНИХ УСКЛАДНЕЛЬ У ПРОЦЕСІ РОЗВ'ЯЗУВАННЯ ЗАДАЧ

Розроблення алгоритму є складною проблемою для студента, яка потребує від нього високого рівня логічного мислення й активної творчої діяльності. Досвід свідчить, що більшість студентів можуть лише розібратися в готовому розв'язку задачі, а самостійно розробити алгоритм здатні лише окремі добре підготовлені студенти.

Одним із можливих виходів з цієї ситуації може бути поступове ускладнення завдань, перехід від простого до складного. Заняття потрібно будувати так, щоб розв'язок однієї задачі вказував шлях до розв'язку іншої, дещо складнішої. Тобто, пропонується метод послідовного уточнення або метод поділу складної задачі на кілька простих.

Спираючись на цей метод, пропонуємо систему взаємозв'язаних задач, що ускладнюються, на конкретному прикладі опрацювання текстових величин. Розв'язання подані алгоритмічною мовою та мовами програмування Turbo Pascal і Бейсік. Система може використовуватися викладачами як методичний розбір задач під час вивчення теми "Опрацювання текстових величин".

Для написання алгоритмів алгоритмічною мовою прийняті такі позначення: функція пошуку номера позиції k першого включення підрядка pdl в рядку s **k:=позиція(pdl,s)**; процедура перетворення величини з рядкового типу (s) в числовий (x) **перетв_чис(s,x)**; процедура перетворення величини з числового типу (x) в рядковий (s) **перетв_ряд(x,s)**.

Приклад 1. Перевірити, чи входить у заданий текст символ "*". Якщо так, то надрукувати символ, що знаходиться безпосередньо перед ним.

Алгоритм розв'язання задачі

```
алг прикл1(літ st,y)
арг st
рез у
поч ціл k
    ввести(st)
    k := позиція("*",st)
    якщо k <> 0
        то якщо k=l то у := " '*' – перший символ"
        інакше у := st[k-1:1]
        все
    інакше у := "в тексті немає символу '*'"
    все
    вивести(у)
кін
```

Програма мовою Паскаль

```
program pr1;
var st: string[255];
    k : integer;
BEGIN
    readln(st);
    k := pos('*',st);
    if k<>0 then if k=1 then
        writeln('* – перший символ')
    else writeln( st[k-1])
    else writeln( 'у тексті немає символу "*" ')
END.
```

Програма мовою Бейсік

```
10 INPUT "введіть рядок :";st$
```

```

20 K= INSTR(ST$,"*")
30 IF K=0 THEN PRINT "у тексті немає
    символу '*'": GOTO 50
40 IF K=1 THEN PRINT "*—перший символ"
    ELSE PRINT MID$(ST$,K-1,1)
50 END

```

Під час створення програми розв'язування цієї задачі слід скористатися тим, що функція pos повертає 0, якщо шуканий підрядок не знайдено, і можливістю індексувати окремі літери рядка, яку надає мова програмування Turbo Pascal.

Змінимо постановку задачі.

Приклад 2. Перевірити, чи входить у заданий текст символ "*". Якщо входить, то надрукувати символ, що знаходиться безпосередньо перед ним, але тільки тоді, коли цей символ є цифра.

Використання для цього попереднього алгоритму є очевидним.

Щоб переконатися, чи є символ цифрою, скористаємося тим, що всі символи впорядковані в кодах ASCII. Якщо символ відповідає умові: ($st[i] \geq '0'$) and ($st[i] \leq '9'$), то можна стверджувати, що цей символ – цифра.

Алгоритм розв'язання задачі

```

алг прикл2( літ st,y)
арг st
рез у
поч ціл k
ввести(st)
k := позиція("(*",st)
якщо k<>0
    то якщо k=1 то у := "*—перший символ"
    інакше якщо st[k-1:l] >="0"
        і st[k-1:l] <="9"
            то у := st[k-1:l]
    інакше у := "символ – не є цифрою"
    все
все
інакше у := "в тексті немає символу '*'"
все
вивести(y)
кін

```

Програма мовою Паскаль

```

program pr2;
var st: string[255];
    k:integer;
BEGIN
readln(st);
k := pos('*',st);
if k <> 0 then if k = 1 then
    writeln('* – перший символ')
    else if (st[k-1] >='0') and (st[k-1] <= '9')
    then writeln(st[k-1])
    else writeln('символ не є цифрою')

```

```

else writeln('у тексті немає символу "*")
END.
```

Програма мовою Бейсік

```

10 INPUT "введіть рядок :";ST$
20 K = INSTR(ST$,"*")
30 IF K=0 THEN PRINT "У тексті немає символу '*'"; GOTO 60
40 IF K=1 THEN PRINT "* – перший символ" : GOTO 60
50 IF MID$(ST$,K-1,1) >= "0" AND MID$(ST$,K-1,1) <= "9" THEN
    PRINT MID$(ST$,K-1,1) ELSE
    PRINT " символ не є цифрою"
60 END
```

Приклад 3. Скористаємося умовою прикладу 2, тільки друкувати будемо символ – цифру, піднісши її значення в квадрат. Единим ускладненням тут є перетворення типу із символьного в числовий, яке здійснюється процедурою **val**.

Алгоритм розв'язування задачі

```

алг прикл3( літ st, ціл x)
арг st
рез x
поч ціл k
k := позиція("?",st)
якщо k <> 0
    то якщо k=1 то вивести (" '?'—перший символ")
    інакше
        якщо st[k-1]>="0" і st[k-1] <= "9"
        то перетв_чис( st[k-1],x)
        x := x*x
        вивести(x)
    інакше вивести(" символ не є цифрою")
    все
    все
інакше вивести(" в тексті немає символу '?')
все
кін
```

Програма мовою Паскаль

```

program pr3;
var st: string[255];
err,x,k : integer;
BEGIN
readln(st);
k := pos('?',st);
if k <> 0 then
    if k = 1 then writeln('*—перший символ')
    else if (st[k-1]>='0') and (st[k-1]<= '9')
        then
        begin
        val(st[k-1],x,err);
        writeln(sqr(x))
```

```

        end
        else writeln('символ не є цифрою')
    else writeln('у тексті немає символу "*")
END.

```

Програма мовою Бейсік

```

10 INPUT "введіть рядок :";st$
20 K=INSTR(ST$,"*")
30 IF K=0 THEN PRINT "У тексті немає символу '*': GOTO 60
40 IF K=1 THEN PRINT "* – перший символ" : GOTO 60
50 IF MID$(ST$,K-1,1)>="0" AND MID$(ST$,K-1,1) <= "9" THEN
    Z$=MID$(ST$,K-1,1) :
    X=VAL(Z$)^2 : PRINT X ELSE
        PRINT" символ не є цифрою"
60 END

```

Приклад 4. Текст заданий наступним чином: $d_1 +(-) d_2 +(-) d_3 +(-) \dots +(-) d_n$, де d_i —цифра.
Знайти значення цієї алгебраїчної суми.

За умовою цифра – лише один символ, за яким іде знак "+" або "-". Тому неважко перетворити кожний другий символ i , перевіряючи знак, що знаходиться в попередній позиції, підрахувати суму, використовуючи стандартний метод її накопичення.

Алгоритм розв'язування задачі

алг прикл4(літ st, ціл sum)

```

арг st
рез sum
поч ціл i,x
ввести(st)
i :=3
перетв_чис(st[ 1:1 ], x)
sum := x
поки i <= довж(st)
    пц
        перетв_чис(st[i:1],x)
        якщо st[i-1:I] = "+"
            то sum := sum + x
        інакше sum := sum-x
        все
        i := i+2
    кц
    вивести(sum)
кін

```

Програма мовою Паскаль

```

program pr4;
var st: string[255];
err,x,i,sum : integer;
BEGIN
readln(st);
i := 3; {позиція наступної цифри}
val(st[1],x,err);
sum := x; {перший доданок}
while i <= length(st) do
begin

```

```

val(st[i],x,err);
if st[i-1] = '+' then sum := sum + x
else sum := sum-x;
i := i + 2;
end;
writeln ('sum=',sum)
END.

```

Програма мовою Бейсік

```

10 INPUT "Введіть рядок :";ST$
20 Z$=MID$(ST$,1,1)
30 X=VAL(Z$) : SUM=X
40 FOR I=3 TO LEN(ST$) STEP 2
50 Z$=MID$(ST$,I,1)
60 X=VAL(Z$)
70 IF MID$(ST$,I-1,1) = "+" THEN
    SUM = SUM + X ELSE SUM= SUM-X
80 NEXT
90 PRINT "SUM=";SUM
100 END

```

Приклад 5. Ускладнимо задачу. Нехай тепер додаватися будуть не лише цифри, а й цілі числа, тобто di не обов'язково займає одну позицію. В цьому випадку потрібно попередньо виділити число з рядка, тобто послідовність символів-цифр, що містяться між символами, які не є цифрами, і тільки після цього перетворювати його до числового типу.

Алгоритм розв'язання задачі

```

алг прикл5(літ st, ціл sum)
арг st
рез sum
поч літ st1, ціл к,x,i
ввести(st)
i :=1
поки st[i:l] >= "0" і st[i:l] <= "9"
    пц
        i:=i+1
    кц
    st1 := st[1:i-1] {перше число}
    перетв_чис(st1,x)
    sum := x
поки i <= довж(st)
    пц
        k :=i
        i:=i+1
    паки (st[i:l]>= "0") і ( st[i:l]<= "9") і (i<=довж(st))
        пц
            i:=i+1
        кц
        st1:=st[k+l:i-(k+l)]
        перетв_чис(st1,x)
        якщо st[k:l] = "+"
            то sum := sum + x
            інакше sum := sum-x

```

```
все
кц
вивести(sum)
кін
```

Програма мовою Паскаль

```
program pr6;
var st: string[255];
    stl: string[6];
    k,x,i,err,sum : integer;
BEGIN
    clrscr;
    write('Введіть рядок'); readln(st);
    i:=1;
    while (st[i] >= '0') and (st[i] <= '9') do i:=i+1;
    stl:=copy(st,1,i-1); {виділяємо перше число}
    val(stl,x,err);
    sum := x; {записуємо в sum перше число}
    while i <= length(st) do
        begin
            k:=i; {запам'ятуємо номер позиції знака}
            i:=i+1;
            while (st[i] >= '0') and (st[i] <= '9') and (i<=length(st)) do i:=i+1;
            stl:=copy(st,k+1,i-(k+1)); {виділяємо наступне число}
            val(stl,x,err);
            if st[k] = '+' then sum :=sum +x else sum := sum-x
        end;
    writeln('sum=',sum)
END.
```

Програма мовою Бейсік

```
10 INPUT " Введіть рядок ":";ST$
20 I = 1
30 WHILE MID$(ST$,I,1) >= "0" AND MID$(ST$,I,1)<= "9"
40 I = I+1
50 WEND
60 ST1$ = MID$(ST$,1,I-1)
70 X= VAL(ST$)
80 SUM = X
90 WHILE I <= LEN(ST$)
100 K = I
110 I = I+1
120 WHILE MID$(ST$,I,1) >= "0" AND MID$(ST$,I,1)<="9" AND I<=LEN(ST$)
130 I = I+1
140 WEND
150 ST1$ = MID$(ST$,K+1,I-K-1)
160 X = VAL(ST1$)
170 IF MID$(ST$,K,1) = "+" THEN
    SUM=SUM + X ELSE SUM=SUM-X
180 WEND
190 PRINT "SUM=";SUM
200 END
```

Приклад 6. Наступне ускладнення: нехай в нашому виразі є знаки не тільки додавання і віднімання, а й множення, тобто значення виразу необхідно підрахувати з дотриманням пріоритету арифметичних операцій. Наприклад: "2+31*4-1".

Найпростіший спосіб розв'язування такої задачі – це виконати спочатку всі операції множення, тобто замінити два числа і знак "*" між ними одним числом, що дорівнює їх добутку. Після чого виконати задачу прикладу 5.

Алгоритм розв'язування задачі

```

алг множ(літ st)
    арг st
    рез st
    поч літ vs1,vs2; ціл k,m1,m2,i,j
    k :=позиція("*", st)
    поки k <> 0
        пц
        i:=k-1
        поки (i >= 1) і (st[i:l] >= "0") і (st[i:l]<= "9")
            пц
            i :=i-1
            кц
        vs1 :=st[i+l:k-i-1]
        перетв_чис( vs1, m1)
        j :=k+1
        поки (j<= довж(st) і (st[j:l] >= "0") і (st[j:l]<="9")
            пц
            j :=j+1
            кц
        vs2 :=st[k+l:j-(k+1)]
        перетв_чис(vs2,m2)
        m1 := m1 * m2
        перетв_ряд(m1, vs1)
        st := st[l:i] + vs1 + st[j-(i+l):довж(st)]
        k :=позиція("*",st)
        кц
    кін
алг приклб(літ st, ціл sum)
    арг st
    рез sum
    поч літ st1, ціл k,x,i
    ввести(st)
    множ { виклик допоміжного алгоритму множення }
    якщо st[1:1] = "-"
        то i := 2
        інакше i := 1
    все
    k :=i
    поки (st[i:l] >= "0") і (st[i:l] <= "9")
        пц
        i :=i+1
        кц
    st1 :=st[k:i-k]
    перетв_чис(st1,x)

```

```

якщо k <> 1
    то sum := -x
    інакше sum := x
все
поки i <= довж(st)
    пц
        k:=i
        i:=i + 1
        поки (st[i:l] >= "0") і (st[i:l] <= "9") і (i <= довж(st))
            пц
                i := i + 1
            кц
            stl := st[k+l:i-(k+l)]
            перетв_чис(stl ,x)
            якщо st[k:l] = "+"
                то sum := sum + x
            інакше sum := sum-x
        все
    кц
вивести(sum)
кін

```

Програма мовою Паскаль

```

program prb;
var s: string[255];
    stl: string[6];
    k,sum,err,x,i: integer;
procedure umnoz;
var k,i,m1,m2,err,j : integer;
    vsl,vs2: string[5];
begin
    k:=pos('*',st);
    while (k <> 0) do
        begin
        i:=k-1;
        while (i >= 1) and (st[i] >='0 ') and (st[i] <='9') do i:=i-1;
        vsl:=copy(st,i+1,k-1-i);
        val(vsl,m1,err); { число, ліворуч від символа "*" }
        j:=k+1;
        while (j<=length(st)) and (st[j]>='0') and (st[j]<='9') do j:=j + 1;
        vs2:=copy(st,k+1,j-(k+1));
        val(vs2,m2,err); { число, праворуч від символа "*" }
        m1:=m1*m2;
        delete(st,i+1,j-(i+1)); { видаляємо обидва числа зі знаком "*" }
        str(m1,vsl); { перетворюємо добуток у рядкову форму }
        insert(vsl,st,i+1); { вставляємо добуток у рядок }
        k:=pos('*',st); { шукаємо наступну операцію множення }
        end;
    end; { umnoz }
BEGIN
clrscr;
write('введіть рядок');
readln(st);

```

```

umnoz; {виклик підпрограми множення}
if st[1] = '-' then i := 2 else i := 1;
{ перевіряємо, чи не від'ємне перше число}
k := i;
while (st[i]>='0') and (st[i]<='9') do i:=i+1;
stl:=copy(st,k,i-k);
val(stl,x,err);
if k <> 1 then sum:=-x else sum :=x; {запис в sum першого числа}
while i <= length(st) do
begin
  k:=i;
  i:=i+1;
  while (st[i]>='0') and (st[i]<='9') and (i<=length(st)) do i:=i+1;
  stl:=copy(st,k+1,i-(k+1));
  val(stl,x,err);
  if st[k] = '+' then sum := sum + x else sum := sum-x
end;
writeln('sum=',sum);
END.

```

Програма мовою Бейсік

```

10 INPUT "рядок : ";ST$
20 ST$=" "+ST$:K=INSTR(ST$,"*")
30 WHILE K<>0
40 I=K-1: PRINT "i=";I
50 WHILE I>=1 AND MID$(ST$,I,1)>="0" AND MID$(ST$,I,1)<="9"
60 I = I-1: WEND
70 VS1$ = MID$(ST$,I+1,K-I-1)
80 M1=VAL(VS1$)
90 J=K+1
100 WHILE J<=LEN(ST$) AND MID$(ST$, J, 1)>="0" AND MID$(ST$,J,1)<="9"
110 J = J+1
120 WEND
130 VS2$ = MID$(ST$,K+1,J-K-1)
140 M2 = VAL(VS2$)
150 M1 = M1*M2 : D = LEN(ST$)
160 VS1$=STR$(M1): ST2$=MID$(ST$,J,D)
170 ST$=MID$(ST$,1,I) + VS1$ + ST2$
180 L=K : K=INSTR(L,ST$,"*")
190 WEND
192 I=1 : WHILE I<=LEN(ST$)
    ' видалення зайвих пропусків
193 Z$=MID$(ST$,I,1): D = LEN(ST$)
194 IF Z$=" " THEN
    ST$=MID$(ST$,1,I-1)+MID$(ST$,I+1,D)
    ELSE I=I+1
195 WEND
200 IF MID$(ST$,1,1) = "-" THEN
    I=2 ELSE I=1
210 K=I
220 WHILE MID$(ST$,I,1)>="0" AND MID$(ST$,I,1)<="9"
230 I = I+1: WEND

```

```
240 ST1$ = MID$(ST$,K,I-K)
250 X=VAL(ST1$)
260 IF K<>1 THEN SUM=-X ELSE SUM=X
270 WHILE I<=LEN(ST$)
280 K = I;I = I+1
290 WHILE MID$(ST$,I,1)>="0" AND MID$(ST$,I,1)<="9" AND I<=LEN(ST$)
300 I=I+1;WEND
310 ST1$ = MID$(ST$,K+1,I-K-1)
320 X = VAL(ST1$)
330 IF MID$(ST$,K,1) = "+" THEN
    SUM =SUM + X ELSE SUM = SUM-X
340 WEND
350 PRINT "sum=";SUM
360 END
```

Отже, використовуючи послідовні ускладнення умови задачі, ми будуємо систему взаємопов'язаних задач, яка розвивається від найпростішого навчального завдання до програмування задач третього рівня складності.

Розв'язування, аналіз і обговорення такої системи підвищують цікавість до предмета і, відповідно, якість навчання.