1. Задания для лексического анализа

языка напишите и протестируйте лексический анализатор.

 $nporpamma \rightarrow (nepemehhbe | npomomun | \phi yhkuus) \{nepemehhbe | npomomun | \phi yhkuus \}$ nеременные \rightarrow **перем** cnucud:mun nерем $\{$;cnucud:mun nерем $\}$ $cnucud \rightarrow uд\{,uд\}$ $mun nepem \rightarrow \mathbf{цел} |\mathbf{вещ}| \mathbf{лог} |\mathbf{симв}| \mathbf{строка} |\mathbf{ничто}| \mathbf{массив} [[eыp] \{, [eыp] \}] mun nepem$ $eup \rightarrow eup_0[?eup_0:eup_0]$ $eup_0 \rightarrow eup_1\{(II|!II|^{\hat{}})!^{\hat{}})eup_1\}$ $eup_1 \rightarrow eup_2\{(\&\&/!\&\&)eup_2\}$ $ewp_2 \rightarrow \{!\}ewp_3$ $6bip_3 \rightarrow 6bip_4 \{(<|>|<=|>=|=|!=)6bip_4 \}$ $eup_4 \rightarrow eup_5 \{(I|~I|~|~|~~)eup_5\}$ $ebip_5 \rightarrow ebip_6 \{(\& | \tilde{\&} | \ll | \gg) ebip_6 \}$ $ebip_6 \rightarrow {^{\sim}}ebip_7$ $выp_7 \rightarrow выp_8\{(+/-)выp_8\}$ $sup_8 \to sup_9 \{ (*//\%/.)sup_9 \}$ $6 \omega p_9 \rightarrow 6 \omega p_{10} [(**/**.) 6 \omega p_9]$ $6bip_{10} \rightarrow [+/-]6bip_{11}$ $6bip_{11} \rightarrow 6bip_{12}$ [#6 bip_{12}] $выp_{12} \to$ целое|вещественное|**истина**|**ложь**|строковое|(выp)|имя $uмя \rightarrow u \pi \{([cnucвыр])|[cnucвыр]\}$ $cnuceup \rightarrow eup\{,eup\}$ $npomomun \to \mathbf{прото}$ заголовок функ заголовок функ $\rightarrow \phi$ унк ид(списформпарам):тип перем $cnuc \phi op mnapa M \rightarrow \varepsilon / spynna napa M \{ : spynna napa M \}$ группа $napam \rightarrow [\mathbf{перем} | \mathbf{конст}] cnucud: mun перем$ ϕ ункция \to заголовок ϕ унк $\{$ тело ϕ унк $\}$ meno функ \rightarrow {nepemenhbe|onepamopbe|} $onepamopы \rightarrow [onepamop]\{; [onepamop]\}$ onepamop
ightarrow ukna | npucваивания | nepexoda | ввода | вывода | условныйусловный \rightarrow если выр то $\{$ mело ϕ унк $\}\{$ инес выр то $\{$ mело ϕ унк $\}\}[$ иначе $\{$ mело ϕ унк $\}\}$ всё пока выр{тело функ}|повторяй{тело функ}покуда выр) $npuceaueahuя \to umя := eыp$ $nepexoda \to \mathbf{выйди} [\mathbf{u}\mathbf{3} \ \mathbf{u}\mathbf{J}] | \mathbf{продолжи} [\mathbf{u}\mathbf{J}] | \mathbf{возврат} [eыp]$ $ввода \rightarrow \mathbf{ввод}(cnucum\ddot{e}H)$ вывода \rightarrow вывод $(cnucum\ddot{e}H)$ $cnucum\ddot{e}H \rightarrow ums\{,ums\}$ Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие регулярные определения): $yeroe \rightarrow decsmurhoe|восьмеричное|deourhoe|mecmhadyamepurhoe$ $\partial e c s m u u ho e \rightarrow \partial e c u u \phi p a ('? \partial e c u u \phi p a)^*$ восьмеричное $\rightarrow 0$ овосьм цифра('?восьм цифра)* $\partial Bouvhoe \rightarrow O(b|B)\partial Bouvh$ цифра('? $\partial Bouvh$ цифра)* $uecmhaduamepuuhoe \rightarrow 0(x|X)uecmh uuppa('?uecmh uuppa)*$ $\partial ec \quad uu\phi pa \rightarrow 0|1|2|3|4|5|6|7|8|9$ восьм иифра $\rightarrow 0/1/2/3/4/5/6/7$ ∂ воичн цифра $\rightarrow 0/1$ $uecm h uu fpa \rightarrow 0 |1|2|3|4|5|6|7|8|9|a|b|c|d|e|f|A|B|C|D|E|F$

Задача 1.1. С помощью генератора лексических анализаторов Мяука для приводимого ниже

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку,

вещественное \to целая часть(дробная часть)?(E|e)(+|-)?порядок

целая_ часть \rightarrow десятичное дробная часть \rightarrow десятичное

 $nopяdok \rightarrow decяmuчнoe$

то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.

Задача 1.2. С помощью генератора лексических анализаторов Мяука для приводимого ниже языка напишите и протестируйте лексический анализатор.

```
nporpamma \rightarrow (nepemehhbe | npomomun | \phi yhkuus ) \{nepemehhbe | npomomun | \phi yhkuus \}
nepeменные \rightarrow var cnucud:mun nepeм\{;cnucud:mun nepeм\}
cnucud \rightarrow uд\{,uд\}
mun nepe M \rightarrow int |float|bool|char|string|void|array|[выр] {, [выр]}|mun nepe M
eup \rightarrow eup_0[?eup_0:eup_0]
eup_0 \rightarrow eup_1\{(|||!|||^{\hat{}})||^{\hat{}}\}
выp_1 \rightarrow выp_2 \{(\&\&/!\&\&)выp_2\}
6bip_2 \rightarrow \{!\}6bip_3
6bip_3 \rightarrow 6bip_4 \{(<|>|<=|>=|=|!=)6bip_4 \}
выp_4 \rightarrow выp_5 \{(I|~I|~/~~)выp_5\}
ebip_5 \rightarrow ebip_6 \{ (\&| \ \&| \ll | \gg) ebip_6 \}
eup_6 \rightarrow \{^{\sim}\}eup_7
eup_7 \rightarrow eup_8\{(+/-)eup_8\}
eup_8 \to eup_9 \{ (*//\%/.)eup_9 \}
6bip_9 \rightarrow 6bip_{10}[(**|**.)6bip_9]
6bp_{10} \rightarrow [+/-]6bp_{11}
ebip_{11} \rightarrow ebip_{12}  [#ebip_{12}]
6ыp_{12} \rightarrowцелое/вещественное/true/false/строковое/(6ыp)/имя
uмя \rightarrow uд\{([cnucвыр])|[cnucвыр]\}
cnuceup \rightarrow eup\{,eup\}
npomomun \rightarrow proto заголовок функ
заголовок функ \rightarrow func ид(списформпарам):mun перем
cnuc \phi op м napa M \rightarrow \varepsilon | группа пapa M \{; группа пapa M \}
rpynna napam \rightarrow [var|const|cnucud:mun nepem
\phiункция \toзаголовок \phiунк \{тело \phiунк\}
meno \phi y + \kappa \rightarrow \{nepemenhue | onepamopu\}
onepamopы \rightarrow [onepamop]\{; [onepamop]\}
onepamop \rightarrow uuкла | npucваивания | nepexoda | ввода | вывода | условный
условный \rightarrowif выр then{meло функ}{elseif выр then{meло функ}}|else{meло функ}|endif
uu\kappa na \rightarrow [ud::](for ud in eup\{meno\_\phi ynk\}|for ud:=eup,eup[,eup[,eup[\}meno\_\phi ynk\}]|
            while \sup\{meno\ \phi y + \kappa\} | \mathbf{repeat}\{meno\ \phi y + \kappa\} | \mathbf{until}\ \mathbf{sup}\}
присваивания \to ums := выр
nepexoda \rightarrow \mathbf{exit} \ [\mathit{ид}] \ [\mathbf{continue} \ [\mathit{ид}] \ [\mathbf{return} \ [\mathit{выр}]
ввода \rightarrowread(списимён)
вывода \rightarrowprint(списимён)
cnucum\ddot{e}H \rightarrow ums\{,ums\}
    Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие ре-
гулярные определения):
uenoe \rightarrow decsmuuhoe|восьмерииное|deouuhoe|шестнадиатерииное
\partial e c s m u ч ho e \rightarrow \partial e c \quad u u \phi p a ('? \partial e c \quad u u \phi p a)^*
восьмеричное \rightarrow 0овосьм цифра('?восьм цифра)*
\partial BOUЧНОЕ \rightarrow O(b|B)\partial BOUЧН цифра('?\partial BOUЧН цифра)*
uecmhaduamepuчhoe \rightarrow 0(x|X)uecmh uuфpa('?uecmh uuфpa)*
\partial ec \quad uu\phi pa \rightarrow 0|1|2|3|4|5|6|7|8|9
```

```
восьм_ цифра \rightarrow 0|1|2|3|4|5|6|7 двоичн_ цифра \rightarrow 0|1 шестн_ цифра \rightarrow 0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f|A|B|C|D|E|F вещественное \rightarrow целая_ часть (дробная_ часть)?(E|e)(+|-)?порядок целая_ часть \rightarrow десятичное дробная_ часть \rightarrow десятичное порядок \rightarrow десятичное
```

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку, то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.

Задача 1.3. С помощью генератора лексических анализаторов Мяука для приводимого ниже языка напишите и протестируйте лексический анализатор.

```
nporpamma \rightarrow (nepemerhue|npomomun|fyyrkuus) \{nepemerhue|npomomun|fyyrkuus\}
nepemehhhee \rightarrow \mathbf{пepem} cnucud:mun nepem\{;cnucud:mun nepem\}
cnucud \rightarrow ид{,ид}
mun nepe \to \mathbf{qел} / \mathbf{вец} / \mathbf{лог} / \mathbf{симв} / \mathbf{строка} / \mathbf{ничто} / \mathbf{массив} / \mathbf{[} / \mathbf{выр} / \mathbf{]} / \mathbf{nun} nepe / \mathbf{nepe} / \mathbf{mun} 
eup \rightarrow eup_0[?eup_0:eup_0]
ewp_0 \rightarrow ewp_1\{(II/!II/^{^{\hspace{-0.5cm} \wedge}}/!^{^{\hspace{-0.5cm} \wedge}})ewp_1\}
\varepsilon \omega p_1 \rightarrow \varepsilon \omega p_2 \{ (\&\& | \&\&) \varepsilon \omega p_2 \}
6bip_2 \rightarrow \{!\}6bip_3
6bip_3 \rightarrow 6bip_4 \{(<|>|<=|>=|=|!=)6bip_4 \}
eup_4 \rightarrow eup_5\{(I|~I|~|~|~~)eup_5\}
eup_5 \rightarrow eup_6 \{(\& | \ \& | \ll | \gg)eup_6 \}
ebip_6 \rightarrow \{^{\sim}\}ebip_7
6bip_7 \rightarrow 6bip_8 \{(+/-)6bip_8\}
sup_8 \to sup_9 \{ (*//\%/.)sup_9 \}
6blp_9 \rightarrow 6blp_{10}[(**|**.)6blp_9]
6bip_{10} \rightarrow [+/-]6bip_{11}
6bip_{11} \rightarrow 6bip_{12} [#6bip_{12}]
6 wp_{12} \to \text{целое} | \text{вещественное} | \textbf{истина} | \textbf{ложь} | \text{строковое} | (6 wp) | uмя
uмя \rightarrow u \pi \{([cnuceup])|[cnuceup]\}
cnuceup \rightarrow eup\{,eup\}
npomomun \to \mathbf{mpoto} заголовок функ
заголовок функ \rightarrow \phiунк ид(списформпарам):тип перем
cnuc \phi op м napa M \rightarrow \varepsilon | spynna napa M \{; spynna napa M \}
группа парам \rightarrow [перем | конст | списид:тип перем
\phiункция \toзаголовок \phiунк \{тело \phiунк\}
meno \phi y + \kappa \rightarrow \{nepemenhue | onepamopu\}
onepamopы \rightarrow [onepamop]\{; [onepamop]\}
onepamop \rightarrow uu\kappa na | npuc ва uвания | nepexoda | ввода | вывода | условный 
условный \rightarrowесли выр \mathbf{ro}{mело функ}{инес выр \mathbf{ro}{mело функ}}|иначе{mело функ}|всё
для ид := выр, выр[, выр] \{ meло функ \} [ завершение \{ meло функ \} []
             пока выр{тело функ}|повторяй{тело функ}покуда выр)
присваивания \to ums := выр
nepexoda \to \mathbf{выйди} [\mathbf{из} \ \mathit{ид}] | \mathbf{продолжи} [\mathit{ид}] | \mathbf{возврат} [\mathit{выр}]
660 \partial a \rightarrow \mathbf{ввод}(cnucum\ddot{e}H)
```

```
вывода →вывод(списимён)
списимён →имя{,имя}
Злесь пелочисленные и вешественные литералы выглялят так (запи
```

Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие регулярные определения):

```
иелое \to десятичное | восьмеричное | двоичное | шестнадцатеричное десятичное \to дес_ цифра ('?дес_ цифра)* восьмеричное \to довосьм_ цифра ('?восьм_ цифра)* двоичное \to двоичное \to двоичн_ цифра ('?двоичн_ цифра)* шестнадцатеричное \to двоичн_ цифра ('?шестн_ цифра)* дес_ цифра \to драго \to двоичн_ цифра \to двоичное дробная_ часть \to десятичное дробная_ часть \to десятичное порядок \to десятичное
```

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку, то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.

Задача 1.4. С помощью генератора лексических анализаторов Мяука для приводимого ниже языка напишите и протестируйте лексический анализатор.

```
nporpa_{MMA} \rightarrow (nepemerhule|npomomun|\phi y + \kappa uu x) \{nepemerhule|npomomun|\phi y + \kappa uu x\}
nepemehhhe \rightarrow var cnucud:mun nepem{;cnucud:mun nepem}
cnucud \rightarrow u_{\mathcal{I}}\{,u_{\mathcal{I}}\}
mun nepe M \rightarrow int |float|bool|char|string|void|array|[выр] {,[выр]}] mun nepe M
eup \rightarrow eup_0[?eup_0:eup_0]
eup_0 \rightarrow eup_1\{(II|!II|^{\hat{}})!^{\hat{}})eup_1\}
eup_1 \rightarrow eup_2 \{(\&\&|!\&\&)eup_2\}
ewp_2 \rightarrow \{!\}ewp_3
6bip_3 \rightarrow 6bip_4 \{(<|>|<=|>=|=|!=)6bip_4 \}
eup_4 \rightarrow eup_5 \{(I | ~I | ^/ ~)eup_5 \}
eup_5 \rightarrow eup_6 \{(\mathbf{k}/\mathbf{k}|\mathbf{k}) \otimes |eup_6\}
ewp_6 \rightarrow \{^{\sim}\}ewp_7
eup_7 \rightarrow eup_8 \{(+/-)eup_8\}
eup_8 \to eup_9 \{ (*//|%//.)eup_9 \}
6bip_9 \rightarrow 6bip_{10} [(**/**.)6bip_9]
6bp_{10} \rightarrow [+/-]6bp_{11}
eup_{11} \rightarrow eup_{12} [\#eup_{12}]
выp_{12} \rightarrow целое|вещественное|true|false|строковое|(выp)|имя
uмя \rightarrow uд\{([cnucвыр])|[cnucвыр]\}
cnuceup \rightarrow eup\{,eup\}
npomomun \rightarrow proto заголовок функ
заголовок функ \rightarrowfunc ид(списформпарам):тип перем
cnuc \phi op м napa M \rightarrow \varepsilon / spynna napa M \{; spynna napa M \}
группа парам \rightarrow [\mathbf{var}|\mathbf{const}]списид:тип перем
\phiункция \toзаголовок \phiунк \{тело \phiунк\}
```

```
onepamopы \rightarrow [onepamop] \{ ; [onepamop] \}
оператор \rightarrow цикла | присваивания | перехода | ввода | вывода | условный
условный \rightarrowif(выр){mело функ}{elif(выр){mело функ}}|else{mело функ}|endif
uu\kappa \lambda a \rightarrow [u\chi:](for u\chi in выр\{me\lambda o \phi y \kappa \}[else\{me\lambda o \phi y \kappa \}]]
            for ид := выр, выр[, выр]{meno  функ}[else{meno  функ}]]
            while \sup\{meno\ \phi y + \kappa\} | \mathbf{repeat}\{meno\ \phi y + \kappa\} | \mathbf{until}\ e \cup p \}
присваивания \to ums := выр
nepexoda \rightarrow \mathbf{exit} \ [ид] \ [continue [ид] \ [return [выр] \ [
ввода \rightarrowread(списимён)
вывода \rightarrowprint(списимён)
cnucum\ddot{e}H \rightarrow ums\{,ums\}
    Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие ре-
гулярные определения):
иелое \rightarrow decsmuvhoe|восьмеричное|deouvhoe|шестнадцатеричное|
\partial e c я m u ч ho e \rightarrow \partial e c \quad u u \phi p a ('? \partial e c \quad u u \phi p a)^*
восьмеричное \rightarrow 0овосьм цифра('?восьм цифра)*
\partial Bouчнoe \rightarrow O(b|B)\partial Bouчн и цифра ('? \partial Bouчн и цифра)*
uecmhaduamepuчhoe \rightarrow 0(x|X)uecmh uuфpa('?uecmh uuфpa)*
\partial ec \quad uu\phi pa \rightarrow 0|1|2|3|4|5|6|7|8|9
восъм цифра \rightarrow 0|1|2|3|4|5|6|7
\partialвоичн цифра \rightarrow 0/1
uecm + uu\phi pa \rightarrow 0|1|2|3|4|5|6|7|8|9|a|b|c|d|e|f|A|B|C|D|E|F
вещественное \to целая часть (дробная часть)?(E|e)(+|-)?nopядок
целая часть \rightarrow \partial e cятичное
дробная часть \rightarrow.десятичное
nopядoк \rightarrow decятичноe
```

 $meno \phi y + \kappa \rightarrow \{nepemenhue | onepamopu\}$

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку, то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.

Задача 1.5. С помощью генератора лексических анализаторов Мяука для приводимого ниже языка напишите и протестируйте лексический анализатор.

```
6bip_{10} \rightarrow [+/-]6bip_{11}
выp_{11} \rightarrow выp_{12}[#выp_{12}]
6 i i p_{12} \rightarrow целое вещественное истина ложь строковое (6 i i p) имя
uмя \rightarrow uд\{([cnucвыр])|[cnucвыр]\}
cnuceup \rightarrow eup\{,eup\}
npomomun \rightarrow \mathbf{прото} заголовок функ
заголовок функ \rightarrow \phiунк ид(списформпарам):тип перем
cnuc \phi op м napa M \rightarrow \varepsilon | группа пapa M \{; группа пapa M \}
группа napam \rightarrow [\mathbf{перем} | \mathbf{конст}] cnucud: mun перем
\phiункция \toзаголовок \phiунк \{тело \phiунк\}
meлo \phi y + \kappa \rightarrow \{nepemehhbe|onepamopы\}
onepamopы \rightarrow [onepamop] \{ ; [onepamop] \}
onepamop \rightarrow uu\kappa \lambda a | npuc ва uвания | nepexoda | ввода | вывода | условный
условный \rightarrowесли выр \mathbf{ro}{mело функ}{инес выр \mathbf{ro}{mело функ}}|иначе{mело функ}|всё
пока выр{тело функ}|повторяй{тело функ}покуда выр)
npucваивания 	oимя (:=|+:=|-:=|*:=|/:=|/:=|**:=|**:=|%:=|&:=|~&:=|<math>\ll:=|\ll:=|\gg:=|I:=|^{\sim}I:=|^{\sim}I:=|^{\sim}I:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|\sim1:=|
                                                    ||:=|!||:=|^^:=|!^^:=|&&:=|!&&:=|
nepexoda \to \mathbf{выйди} [\mathbf{us} \ ud] | \mathbf{продолжи} [\mathbf{ud}] | \mathbf{возврат} [eыp]
660 \partial a \rightarrow \mathbf{ввод}(cnucum\ddot{e}H)
выводa \rightarrowвывод(cnucum\ddot{e}_H)
cnucum\ddot{e}H \rightarrow ums\{,ums\}
       Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие ре-
гулярные определения):
uenoe \rightarrow decsmuuhoe|восьмеричное|deouuhoe|mecmhaduamepuuhoe
\partial e c s m u u ho e \rightarrow \partial e c u u \phi p a ('? \partial e c u u \phi p a)^*
восьмеричное \rightarrow 0овосьм цифра('?восьм цифра)*
dвоичное \rightarrow 0(b|B)dвоичн цифра ('?dвоичн цифра)*
wecmhaduamepuчное \rightarrow 0(x|X)wecmh uuфpa('?wecmh uuфpa)*
\partial ec \quad uu \phi pa \rightarrow 0/1/2/3/4/5/6/7/8/9
восьм иифра \rightarrow 0/1/2/3/4/5/6/7
\partialвоичн ци\phiра \rightarrow 0/1
uecm h uu fpa \rightarrow 0 |1|2|3|4|5|6|7|8|9|a|b|c|d|e|f|A|B|C|D|E|F
вещественное \to целая часть (дробная часть)?(E|e)(+|-)?порядок
иелая часть \rightarrow \partial e cятичное
дробная часть \rightarrow.десятичное
nopядoк \rightarrow decяmuчнoe
```

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку, то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.

Задача 1.6. *С* помощью генератора лексических анализаторов Мяука для приводимого ниже языка напишите и протестируйте лексический анализатор.

```
программа \rightarrow (переменные | прототип | функция) { переменные | прототип | функция } переменные \rightarrow var списид: mun_nepem {; cnucud:mun_nepem } cnucud \rightarrow uд {, ид } mun_nepem \rightarrow int | float | bool | char | string | void | array[[выр] {, [выр]}] | mun_nepem \rightarrow выр \rightarrow выр o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o
```

```
eup_0 \rightarrow eup_1\{(|||!|||^{^}|!^{^})eup_1\}
sup_1 \rightarrow sup_2 \{(\&\&/!\&\&)sup_2\}
6bip_2 \rightarrow \{!\}6bip_3
6bip_3 \rightarrow 6bip_4 \{(<|>|<=|>=|=|!=)6bip_4 \}
ebip_4 \rightarrow ebip_5 \{(I | ~I | ~I | ~I \sim )ebip_5 \}
eup_5 \rightarrow eup_6 \{(\& | \tilde{\&} | \ll | \gg) eup_6 \}
eup_6 \rightarrow \{^{\sim}\}eup_7
eup_7 \rightarrow eup_8\{(+/-)eup_8\}
eup_8 \rightarrow eup_9\{(*//\%/.)eup_9\}
6blp_9 \rightarrow 6blp_{10}[(**/**.)6blp_9]
6bip_{10} \rightarrow [+/-]6bip_{11}
6bip_{11} \rightarrow 6bip_{12} [#6bip_{12}]
6ыp_{12} \rightarrow целое/вещественное/true/false/строковое/(6ыp)/имя
uмя \rightarrow uд\{([cnucвыр])|[cnucвыр]\}
cnuceup \rightarrow eup\{,eup\}
npomomun \rightarrow proto заголовок функ
заголовок функ \rightarrow func ид(списформпарам):тип перем
cnuc \phi op м napa M \rightarrow \varepsilon | spynna napa M \{; spynna napa M \}
rpynna napam \rightarrow [var|const|cnucud:mun nepem
\phiункция \toзаголовок \phiунк \{тело \phiунк\}
meno \phi y \mu \kappa \rightarrow \{nepemenhue | onepamopu\}
onepamopы \rightarrow [onepamop]\{; [onepamop]\}
оператор \rightarrow цикла | присваивания | перехода | ввода | вывода | условный
условный \rightarrow if(выр){mело_функ}{elif(выр){mело_функ}}|else{mело_функ}|endif
uu\kappa \lambda a \rightarrow [u\chi:](for u\chi in выр\{me\lambda o \phi y \kappa \}[else\{me\lambda o \phi y \kappa \}]]
           for ид := выр, выр[,выр]{mело функ}[else{mело функ}]]
           while выр{тело функ}|repeat{тело функ}until выр)
npucваивания 	oимя (:=|+:=|-:=|*:=|/:=|/.:=|**:=|**.:=|%:=|%:=|~&:=|%:=|%:=||:=|^-!=|^-:=|
                           ||:=|!||:=|^^:=|!^^:=|&&:=|!&&:=|
nepexoda \rightarrow exit [ид]/continue [ид]/return [выр]
ввода \rightarrowread(списимён)
вывода \rightarrowprint(списимён)
cnucum\ddot{e}H \rightarrow ums\{,ums\}
    Здесь целочисленные и вещественные литералы выглядят так (записаны соответствующие ре-
гулярные определения):
uenoe \rightarrow decsmuuhoe|восьмеричное|deouuhoe|mecmhaduamepuuhoe
\partial ecsmuчнoe \rightarrow \partial ec цифра('?\partial ec цифра)*
восьмеричное \rightarrow 0овосьм цифра('?восьм цифра)*
\partial BOUЧНОЕ \rightarrow O(b|B)\partial BOUЧН цифра('?\partial BOUЧН цифра)*
uecmhaduamepuчhoe \rightarrow 0(x|X)uecmh uuфpa('?uecmh uuфpa)*
\partial ec \quad uu\phi pa \rightarrow 0|1|2|3|4|5|6|7|8|9
восьм_ цифра 
ightarrow 0|1|2|3|4|5|6|7
\partialвоичн цифра \rightarrow 0/1
uecm H uu фра \rightarrow 0/1/2/3/4/5/6/7/8/9/a/b/c/d/e/f/A/B/C/D/E/F
вещественное \toцелая часть(дробная часть)?(E|e)(+|-)?порядок
целая часть \rightarrow \partial e cятичное
дробная часть \rightarrow.десятичное
nopядoк \rightarrow decяmичноe
```

Строковый литерал — это любая (в том числе и пустая) последовательность символов, заключённая в одинарные кавычки. Если в строковом литерале нужно указать саму одинарную кавычку, то эту кавычку нужно удвоить. Символьный литерал представляет собой либо заключённый в одинарные кавычки символ, либо символ, заданный своим кодом. Последнее представляет собой знак \$, после которого идёт целое число, являющееся кодом символа. Число это может быть двоичным, восьмеричным, шестнадцатеричным, или десятичным Внутреннее представление строковых и символьных литералов — в кодироке Unicode, а именно, в её четырёхбайтовом варианте. Текст программы — только в кодировке UTF-8.

Идентификатор представляет собой последовательность букв, цифр, и знаков подчёркивания, и может начинаться либо с буквы, либо со знака подчёркивания. Под буквой понимается либо русская буква, либо латинская буква.

При реализации сканера для хранения целочисленного значения лексемы следует использовать значение типа unsigned __int128, а для хранения вещественного — значение типа __float128.