

**PRYWATNA WYŻSZA SZKOŁA NAUK
SPOŁECZNYCH, KOMPUTEROWYCH I MEDYCZNYCH**

**WYDZIAŁ NAUK SPOŁECZNYCH I
TECHNIK KOMPUTEROWYCH**

**Ćwiczenie
z programowania niskopoziomowego**

„Obsługa klawiatury i myszy”

Wariant N 8

Opracował

Grzegorz Makowski

III rok Informatyki
Studia niestacjonarne

Prowadzący
Prof. dr hab. inż. Aleksandr Timofiejew

Warszawa 2019/2020

Spis treści

| | |
|-----------------------------|----|
| Zadanie a | 3 |
| Tworzenie okna konsolowego. | 3 |
| Opracowanie zadania | 3 |
| Testowanie | 3 |
| Zadanie b | 5 |
| Klawiatura | 5 |
| Opracowanie zadania | 5 |
| Testowanie | 5 |
| Zadanie c | 7 |
| Mysz | 7 |
| Opracowanie zadania | 7 |
| Testowanie | 18 |

Zadanie a

Tworzenie okna konsolowego.

Napisać program, w którym otworzyć nowe okno konsolowe.

Wyświetlić okno i komunikaty innymi kolorami niż kolory okna bazowego.

Na środku okna wypisać komunikat „Kolory są ustawione przez Grzegorz Makowski”.

Ponieważ po uruchomieniu aplikacji okno szybko zniknie, dodać do programu cykl – opóźnienie:


```
mov ECX, 07FFFFFFFh
```

```
etyk: loop etyk ; opóźnienie zamknięcia okna
```

Opracowanie zadania

;Kod programu w zadaniu C

Testowanie



```
PWSBiA - Grzegorz Makowski@2020

Warszawska Uczelnia Medyczna
im. Tadeusza Koźłuka

Ćwiczenie z programowania niskopozomowego

Obsługa klawiatury i myszy
Wariant N 8

"Obsługa klawiszy H i I oraz Lewy Alt + lewy myszy"

.....: MENU :.....

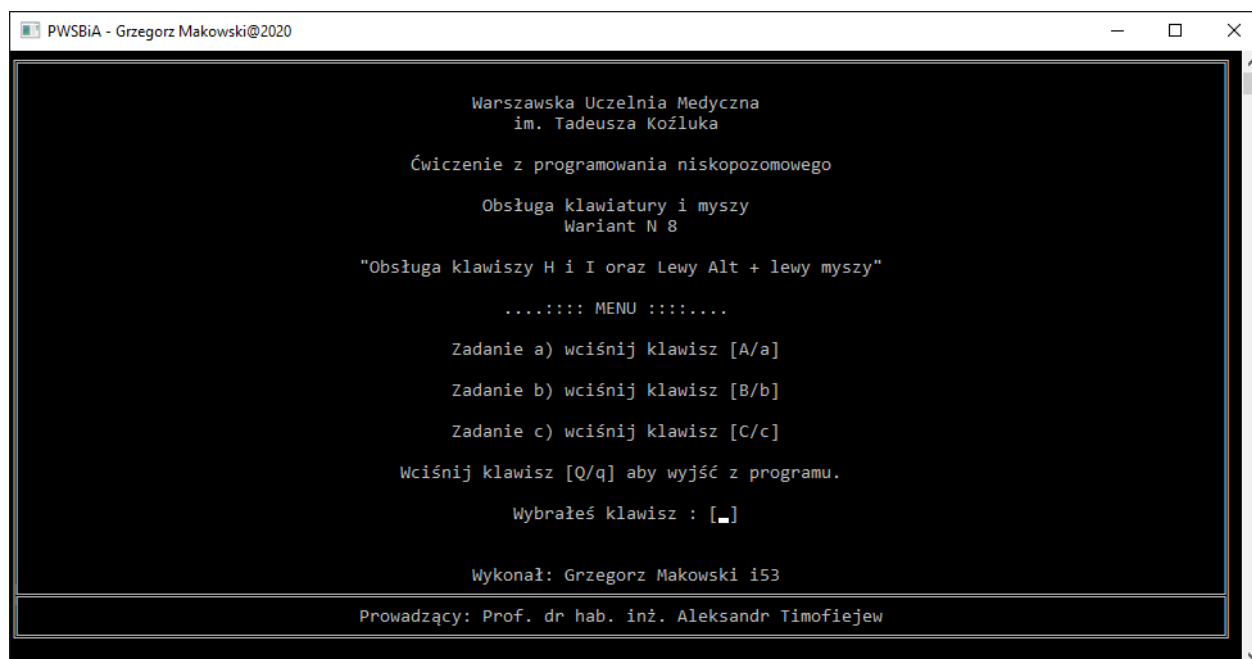
Zadanie a) wciśnij klawisz [A/a]
Zadanie b) wciśnij klawisz [B/b]
Zadanie c) wciśnij klawisz [C/c]

Wciśnij klawisz [Q/q] aby wyjść z programu.

Wybrałeś klawisz : [ ]

Wykonał: Grzegorz Makowski i53

Prowadzący: Prof. dr hab. inż. Aleksandr Timofiejew
```



Zadanie b

Klawiatura

Dodać do programu reakcję na naciśnięcie przycisku.

Przyjmować znaki z klawiatury, analizować i wychodzić z aplikacji (zamykać okno) w przypadku kolejnego naciśnięcia przycisków H , i lub h, i.

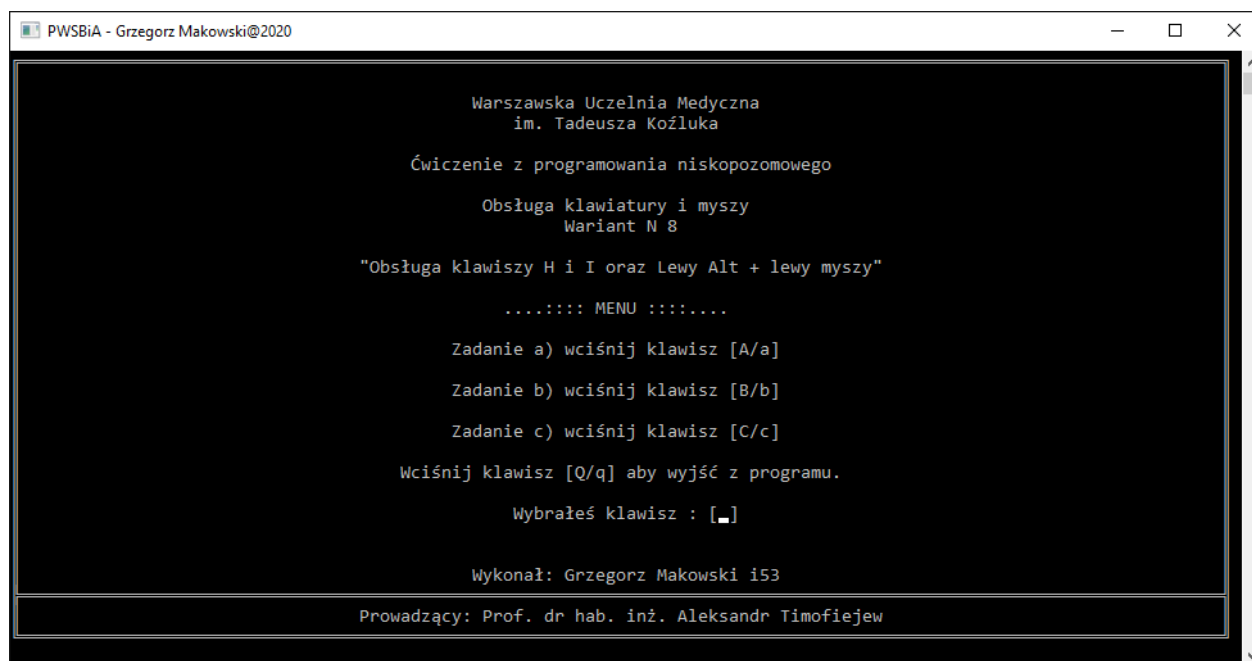
Na dole okna wypisać komunikat o tym, jak zakończyć aplikację.

Cykl – opóźnienie zablokować.

Opracowanie zadania

;Kod programu w zadaniu C

Testowanie



```
PWSBiA - Grzegorz Makowski@2020

Warszawska Uczelnia Medyczna
im. Tadeusza Kościuszki

Ćwiczenie z programowania niskopoziomowego

Obsługa klawiatury i myszy
Wariant N 8

"Obsługa klawiszy H i I oraz Lewy Alt + lewy myszy"

.....: MENU :.....

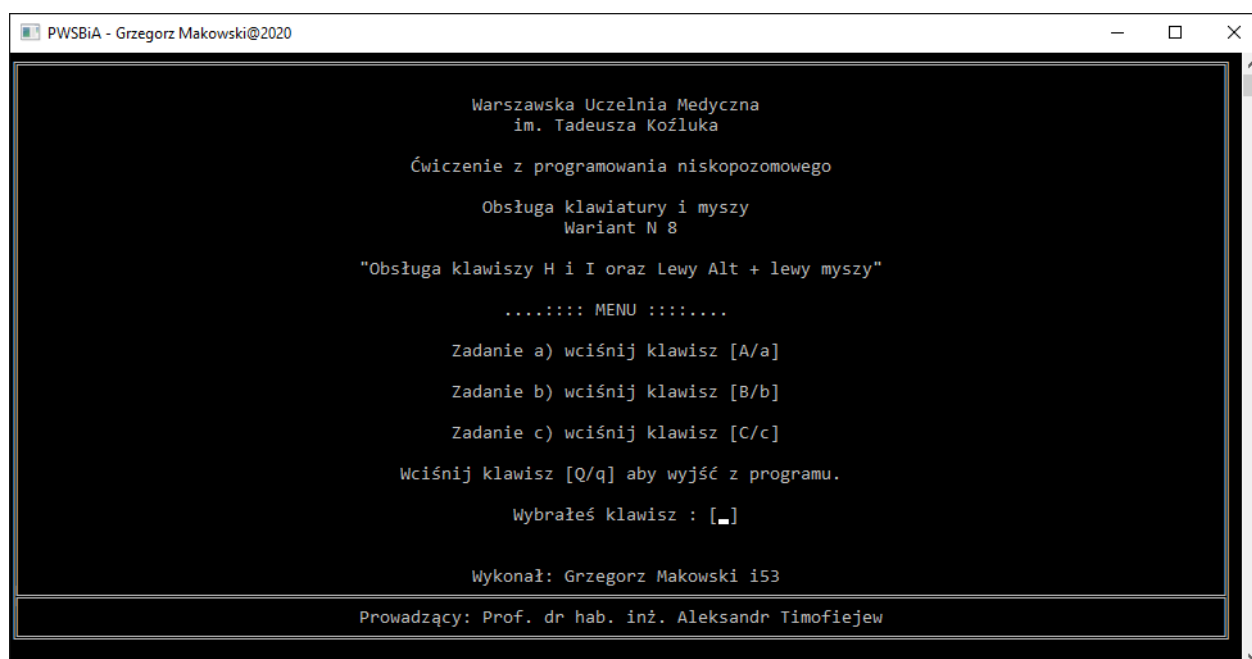
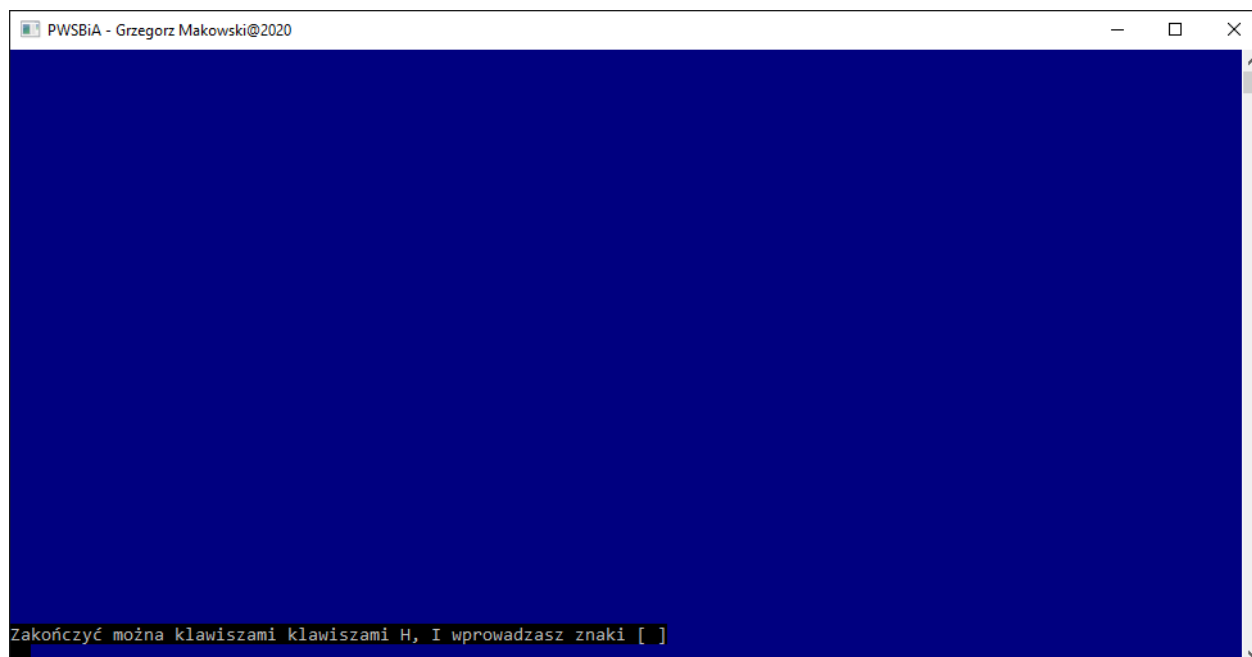
Zadanie a) wciśnij klawisz [A/a]
Zadanie b) wciśnij klawisz [B/b]
Zadanie c) wciśnij klawisz [C/c]

Wciśnij klawisz [Q/q] aby wyjść z programu.

Wybrałeś klawisz : [ ]

Wykonał: Grzegorz Makowski i53

Prowadzący: Prof. dr hab. inż. Aleksandr Timofiejew
```



Zadanie c

Mysz

Dodać do programu reakcję na naciśnięcie okna myszą.

Jednokrotne naciśnięcie lewym przyciskiem myszy musi przemieszczać kursor.

Od miejsca kursora muszą być wyświetlane wprowadzane znaki.

Wychodzić z aplikacji (zamykać okno) po kombinacji naciśnięć przycisku i myszy według reguły Lewy Alt i Lewy klawisz myszy.

Do komunikatu na dole okna dodać informację o tym, jak zakończyć aplikację za pomocą myszy.

Opracowanie zadania

```
-----  
;plik mojemakra.mac  
-----
```

```
-----  
;      mojemakra.mac      |  
;      ver 1.0            |  
;      Autor: Grzegorz Makowski |  
;      MASM ver: 6.14.8444 |  
;      ost. akt. 27.05.2020 |  
-----
```

```
-----  
;      Makro do deskryptorow      |  
-----
```

```
podajdeskr macro handle, deskrypt  
push handle  
call GetStdHandle  
mov deskrypt, eax ; deskryptor bufora konsoli  
endm
```

```
-----  
;      Konwersja polskich znakow      |  
;      Przyjmuje tab znakowa i bufor      |  
-----
```

```
plznaki macro text, bufor  
invoke CharToOemA, addr text, addr bufor  
endm
```

```
-----  
;Wyświetlanie wyniku- tekstu na ekranie      |  
; bufor z plznaki lub kazy inny tekst      |  
-----
```

```
wyswietl macro bufor, rozmiar  
push 0 ; rezerwa, musi być zero  
push offset rout ; wskaźnik na faktyczną ilość wyprowadzonych znaków  
push rozmiar ; ilość znaków  
push offset bufor ; wskaźnik na tekst w buforze  
push hout ; deskryptor buforu konsoli  
call WriteConsoleA ; wywołanie funkcji WriteConsoleA  
endm
```

```
-----  
;      Makro do robienia odstepu linii      |  
;przyjmuje kod hex nowej linni i ilosc zn.      |  
-----
```

```
nowalinia macro nowa, ilznk  
push 0 ; rezerwa, musi być zero  
push offset rout ; wskaźnik na faktyczną ilość wyprowadzonych znaków  
push ilznk ; ilość znaków  
push offset nowa ; wskaźnik na tekst  
push hout ; deskryptor buforu konsoli  
call WriteConsoleA ; wywołanie funkcji WriteConsoleA  
endm
```

```

;-----|
;      Wyszewlenie nagłówka      |
;-----|
Naglowek macro
;--- wywołanie funkcji GetStdHandle
podajdeskr STD_OUTPUT_HANDLE, hout
plznaki naglow, buf ; konwersja
;--- wyszewlenie powitania -----
wyswietl buf, rozmN ; wyszewlenie
nowalinia nl, 2 ; nowa linia
nowalinia nl, 2 ; nowa linia
endm
;-----|
;      funkcja drukuj znak      |
;-----|

```

```

drukujZnak macro znak, ile
mov bl,1
@@:
cmp bl,ile
je @F
wyswietl znak, 1
add bl,1
jmp @B
@@:
endm

```

```

plik cw8.asm

```

```

.586p
.model flat, stdcall
;-----|
;      wczytanie plikow zewnetrznych      |
;-----|
;      wczytanie wasnych makr z pliku      |
;-----|
include mojemakra.mac ; Makra
;-----|
;      biblioteki systemowe i masm      |
;-----|
includelib .lib\user32.lib
includelib .lib\kernel32.lib
includelib .lib\masm32.lib
;-----|
;      Stałe do zadania 8      |
;-----|
KEY_EVENT EQU 1h ; zdarzenie klawiatury
MOUSE_EVENT EQU 2h ; zdarzenie myszy
MENU_EVENT EQU 8h
FOCUS_EVENT EQU 10h
RIGHT_ALT_PRESSED EQU 1h ; Alt prawy
LEFT_ALT_PRESSED EQU 2h ; Alt lewy
RIGHT_CTRL_PRESSED EQU 4h ; Ctrl prawy
LEFT_CTRL_PRESSED EQU 8h ; Ctrl lewy
SHIFT_PRESSED EQU 10h ; Shift naciśnięty
NUMLOCK_ON EQU 20h ; Num Lock włączony
SCROLLLOCK_ON EQU 40h ; Scroll Lock włączony
CAPSLOCK_ON EQU 80h ; Caps Lock włączony
ENHANCED_KEY EQU 100h ; klawisz jest zwolniony
FROM_LEFT_1ST_BUTTON_PRESSED EQU 1h ; przycisk lewy
RIGHTMOST_BUTTON_PRESSED EQU 2h ; przycisk prawy
FROM_LEFT_2ND_BUTTON_PRESSED EQU 4h ; przycisk srodkowy
FROM_LEFT_3RD_BUTTON_PRESSED EQU 8h ; przycisk trzeci z lewa
FROM_LEFT_4ND_BUTTON_PRESSED EQU 10h ; przycisk 4-ty z lewa
MOUSE_MOVED EQU 1h ; mysz przesuwa się
DOUBLE_CLICK EQU 2h ; podwójne kliknięcie
MOUSE_WHEELED EQU 4h ; poruszenie kółkiem myszy
;--- struktury z pliku windows.inc -----
coord STRUCT
X DW ?
Y DW ?
coord ENDS
MOUSE_EVENT_RECORD STRUCT
dwMousePosition coord <> ; współrzędne X, Y kursora myszy

```



```

dwButtonState dword ? ; znaczniki naciśnięcia przycisków myszy
dwControlKeyState dword ? ; znaczniki naciśnięcia klawiszy sterujących
dwEventFlags dword ? ; znaczniki przesuwania i podwójnego klik-niecie myszy
MOUSE_EVENT_RECORD ENDS
KEY_EVENT_RECORD STRUCT
bKeyDown dword ? ;znacznik naciśnięcia któregośkolwiek klawisza
wRepeatCount WORD ? ; ilość powtórzeń kodu przy długim naciśnięciu
wVirtualKeyCode WORD ? ; wirtualny kod klawisza
wVirtualScanCode WORD ? ; scan-kod klawisza
UNION
UnicodeChar WORD ? ; UNICODE kod klawisza
AsciiChar BYTE ? ; ASCII kod klawisza
ENDS
dwControlKeyState dword ? ;znaczniki klawiszy sterujących
KEY_EVENT_RECORD ENDS
WINDOW_BUFFER_SIZE_RECORD STRUCT
dwSize coord <>
WINDOW_BUFFER_SIZE_RECORD ENDS
MENU_EVENT_RECORD STRUCT
dwCommandId dword ?
MENU_EVENT_RECORD ENDS
FOCUS_EVENT_RECORD STRUCT
bSetFocus dword ?
FOCUS_EVENT_RECORD ENDS
INPUT_RECORD STRUCT ;struktura z informacją o zdarzeniu
EventType WORD ? ; typ zdarzenia
two_byte_alignment WORD ? ; wyrównanie do granicy/4
UNION
KeyEvent KEY_EVENT_RECORD <>
MouseEvent MOUSE_EVENT_RECORD <>
WindowBufferSizeEvent WINDOW_BUFFER_SIZE_RECORD <>
MenuEvent MENU_EVENT_RECORD <>
FocusEvent FOCUS_EVENT_RECORD <>
ENDS
INPUT_RECORD ENDS
;-----|
;   stale z pliku .\include\windows.inc   |
;-----|
STD_INPUT_HANDLE equ -10
STD_OUTPUT_HANDLE equ -11
GetStdHandle proto :dword
ReadConsoleA proto :dword,:dword,:dword,:dword,:dword
WriteConsoleA proto :dword,:dword,:dword,:dword,:dword
ExitProcess proto :dword
wsprintfA proto c :vararg
lstrlenA proto :dword
;--- funkcje API Win32 z pliku .\include\user32.inc ---
CharToOemA proto :dword,:dword
AllocConsole proto
FreeConsole proto
SetConsoleTitleA proto :dword
SetConsoleTextAttribute proto :dword,:dword
GetLargestConsoleWindowSize proto :dword
FillConsoleOutputAttribute proto :dword,:dword,:dword,:coord,:dword
SetConsoleCursorPosition proto :dword, :coord
ReadConsoleInputA proto :dword,:dword,:dword,:dword
FlushConsoleInputBuffer proto :dword
WriteConsoleOutputCharacterA proto :dword,:dword,:dword,:coord,:dword

;-----|
;--- własne defenicje   |
;-----|
Cls proto ; czyszczenie ekranu przez przewinięcie
Menu proto ; wyświetlanie menu
;-----|
; --- koniec własnych def   |
;-----|
;-----|
;   stale   |
;-----|
mbuf = 512
kolorb = 00010001b ; Niebieski - BG_I,G,R,B,FG_I,G,R,B
kolorw = 11111111b ; Biały - BG_I,G,R,B,FG_I,G,R,B
kolorc = 01000100b ; Czerwony BG_I,G,R,B,FG_I,G,R,B
kolor1 = 00000111b ; Kolory: BG_I,G,R,B,FG_I,G,R,B

```

```

kolor2 = 01100001b ; Kolory: BG_I,R,G,B,FG__I,R,G,B
;-----
;clear db "20h"
_data segment
hout dd 0
hinp dd ? ; deskryptor buforu wejściowego
rout dd 0
szkola1 db "Warszawska Uczelnia Medyczna",0
sz1 dd $ - szkoła1
align 4
szkola2 db "im. Tadeusza Koźłuka",0
sz2 dd $ - szkoła2
align 4
szkola3 db "Ćwiczenie z programowania niskopozomowego",0
sz3 dd $ - szkoła3
align 4
szkola4 db "Obsługa klawiatury i myszy",0
sz4 dd $ - szkoła4
align 4
szkola5 db "Wariant N 8",0
sz5 dd $ - szkoła5
align 4
szkola6 db "Obsługa klawiszy H i I oraz Lewy Alt + lewy myszy",0
sz6 dd $ - szkoła6
align 4
tytul db "PWSBiA - Grzegorz Makowski@2020",0
ALIGN 4
tytulr dd $ - tytul ;ilość znaków tablicy tytul
tytulA db "Aplikację konsolową stworzył Grzegorz Makowski",0
ALIGN 4
tytulrozm dd $ - tytulA ;ilość znaków tablicy tytul
align 4
menuTekst db "..... MENU :..... "
menuR dd $ - menuTekst
naglow db "Autor aplikacji Grzegorz Makowski i53",0
align 4 ; przesuniecie do adresu podzielonego na 4
rozmN dd $ - naglow ;ilość znaków w tablicy
zadanieA db "Zadanie a) wciśnij klawisz [A/a]",0
rozmA dd $ - zadanieA ; ilosc znakow tekstu zadanieA
align 4
zadanieB db "Zadanie b) wciśnij klawisz [B/b]",0
rozmB dd $ - zadanieB
align 4
zadanieC db "Zadanie c) wciśnij klawisz [C/c]",0
rozmC dd $ - zadanieC
align 4
zakonczenie db "Wciśnij klawisz [Q/q] aby wyjść z programu.",0
zakR dd $ - zakonczenie
align 4
wybrany db "Wybrałeś klawisz : [ ]",0
wybrR dd $ - wybrany
align 4
student db "Wykonał: Grzegorz Makowski i53",0
stuR dd $ - student
align 4
profesor db "Prowadzący: Prof. dr hab. inż. Aleksandr Timofiejew",0
proR dd $ - profesor
align 4
tab dd 100 dup(0)
bufor db mbuf dup(0)
leng dd 0
buf db mbuf dup(0) ; bufor pomocniczy
buff db mbuf dup(0) ; bufor pomocniczy
zmMenudd 4 ; zmienna do trzymania wyniku z manu
rbuf dd mbuf
rinp dd 0 ;faktyczna ilość wprowadzonych znaków
ile dd 0
nl db 0Dh, 0Ah, 0; nowa linia
nl2 db 0Dh,0Ah,20h,0 ; nowa inne formatowanie
nxt db 13,10,0 ; następny wiersz
rogGL db 0c9h ; rog góra lewy
rogGP db 0bbh ; róg góra prawy
prosta db 0cdh ; prosta pozioma
pion db 0bah ; prosta pionowa
pionL db 0cch ; pionowa w lewo skrzyzowanie
pionP db 0b9h ; pionowa w prawo skrzyzowanie

```

```

rogDL db 0c8h; róg dół lewy
rogDP db 0bch ; róg dół prawy
spacja db 020h
licznik1 db "1",0
licznik2 db "15"
;-----
nzdarz dd ? ; ilość zdarzeń
zapis INPUT_RECORD <>
YX COORD <>
symb db 'A'
rkom dd ? ; ilość komórek konsoli
rfakt dd ? ; faktyczna ilość komórek konsoli
wierszB db "Zakończyć można klawiszami klawiszami H, I wprowadzasz znaki [ ]",0Dh,0Ah,0
ALIGN 4
wierszrB dd $ - wierszB ;ilość znaków tablicy wiersz
niebieski dd kolorb
biały dd kolorw
czerwony dd kolorc
atryb dd kolor2
atryb1 dd kolor1
align 4
tekst db "Kolory są ustawione przez Grzegorza Makowskiego"
rozmT dd $ - tekst
align 4
wierszC db "Zadanie można zakończyć klawiszami Lewy Alt + Lewy przycisk myszy lub "
db " klawiszami H, I lub h, i",0Dh,0Ah,0
wierszrC dd $ - wierszC ;ilość znaków tablicy wiersza
align 4
LeKIMyszy db "FROM_LEFT_1ST_BUTTON_PRESSED",0
LeKIMyszyr dd $ - LeKIMyszy
align 4
;-----
_data ends
;-----|
; Koniec segmentu danych |
;-----|

;-----|
; Początek segmentu kodu |
;-----|
_text segment
start:
;--- tworzenie konsoli ---
INVOKE FreeConsole
INVOKE AllocConsole
;--- nagłówek okna ---
INVOKE CharToOemA,OFFSET tytuł,OFFSET tytuł ; konwersja polskich znaków
INVOKE SetConsoleTitleA,OFFSET tytuł
;--- deskryptory buforów ---
INVOKE GetStdHandle, STD_OUTPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hout, EAX
INVOKE GetStdHandle, STD_INPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hinp, EAX
;--- obliczenie rozmiaru buforu ekranu
INVOKE GetLargestConsoleWindowSize,hout
mov YX, EAX
movzx EAX, YX.X
movzx EBX, YX.Y
mul EBX
mov rkom, EAX
mov YX.X, 0
mov YX.Y, 0
call Menu
mov YX.X, 69
mov YX.Y, 22
INVOKE SetConsoleCursorPosition,hout,COORD PTR [YX] ;ustawienie pozycji kursora
;--- pętla
powt:
;--- odczyt komunikatu zdarzenia
INVOKE FlushConsoleInputBuffer,hinp ;czyszczenie kolejki komunikatów
INVOKE ReadConsoleInputA,hinp,OFFSET zapis,1,OFFSET nzdarz ; odczyt komunikatu zdarzenia
cmp WORD PTR [zapis.EventType], KEY_EVENT
je @F
jmp powt
@@:
jmp klaw

```

```

;--- komunikat od klawiatury ---
klaw:
cmp (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).bKeyDown, 0 ; porównanie z zerem
jne @F
jmp powt ; jeśli nie ma naciśnięcia
@@:
mov AL,(KEY_EVENT_RECORD PTR [zapis.KeyEvent]).ASCIIChar
mov symb,AL
cmp AL, 051h ; porównanie z kodem Q
jne @F
jmp kon
@@:
cmp AL, 071h ; porównanie z kodem q
jne @F
jmp kon
@@:
cmp AL, 041h ; porównanie z kodem A
jne @F
jmp zadanie_a
@@:
cmp AL, 061h ; porównanie z kodem a
jne @F
jmp zadanie_a
@@:
cmp AL, 042h ; porównanie z kodem B
jne @F
jmp zadanie_b
@@:
cmp AL, 062h ; porównanie z kodem b
jne @F
jmp zadanie_b
@@:
cmp AL, 043h ; porównanie z kodem C
jne @F
jmp zadanie_c
@@:
cmp AL, 063h ; porównanie z kodem c
jne @F
jmp zadanie_c
@@:
INVOKE WriteConsoleOutputCharacterA,hout,OFFSET symb,1,COORD PTR [YX],OFFSET rout ; wypisywanie znaku
jmp powt
;-----|
; koniec menu |
;-----|
;-----|
; Zadanie a |
;-----|
zadanie_a:
;--- tworzenie konsoli ---
INVOKE FreeConsole
INVOKE AllocConsole
;--- nagłówek okna ---
INVOKE CharToOemA,OFFSET tytul,OFFSET tytul ; konwersja polskich znaków
INVOKE SetConsoleTitleA,OFFSET tytul
;--- deskryptory buforów ---
INVOKE GetStdHandle, STD_OUTPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hout, EAX
INVOKE GetStdHandle, STD_INPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hinp, EAX
;--- obliczenie rozmiaru buforu ekranu
INVOKE GetLargestConsoleWindowSize,hout
mov YX, EAX
movzx EAX, YX.X
movzx EBX, YX.Y
mul EBX
mov rkom, EAX
;--- napęlnienie komórek jednakowym atrybutem ---
mov YX.X, 0
mov YX.Y, 0
;--- ustawienie koloru konsoli ---
INVOKE SetConsoleTextAttribute,hout,kolor1
INVOKE FillConsoleOutputAttribute, hout, kolor2, rkom, COORD PTR [YX], OFFSET rfakt
;--- wyprowadzenie komunikatu informacyjnego ---
mov YX.X, 10

```

```

mov YX.Y, 15
INVOKE SetConsoleCursorPosition, hout, YX
plznaki tekst, tekst ; makro
wyswietl offset tekst, rozmT ; makro
;--- opóźnienie zamknięcia okna ---
mov ECX, 07FFFFFFh
etykA: loop etykA
;--- zamknięcie konsoli
konA:
INVOKE FreeConsole
jmp start
;-----|
; koniec zadania a |
;-----|
;-----|
; Zadanie b |
;-----|
zadanie_b:
;--- tworzenie konsoli ---
INVOKE FreeConsole
INVOKE AllocConsole
;--- nagłówek okna ---
INVOKE CharToOemA, OFFSET tytuł, OFFSET tytuł ; konwersja polskich znaków
INVOKE SetConsoleTitleA, OFFSET tytuł
;--- deskryptory buforów ---
INVOKE GetStdHandle, STD_OUTPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hout, EAX
INVOKE GetStdHandle, STD_INPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hinp, EAX
;--- obliczenie rozmiaru buforu ekranu
INVOKE GetLargestConsoleWindowSize, hout
mov YX, EAX
movzx EAX, YX.X
movzx EBX, YX.Y
mul EBX
mov rkom, EAX
;--- napełnienie komórek jednakowym atrybutem ---
mov YX.X, 0
mov YX.Y, 0
INVOKE FillConsoleOutputAttribute, hout, niebieski, rkom, \
COORD PTR [YX], OFFSET rfakt
;--- wyprowadzenie komunikatu informacyjnego ---
mov YX.X, 0
mov YX.Y, 29
INVOKE SetConsoleCursorPosition, hout, YX
INVOKE SetConsoleTextAttribute, hout, atryb1
plznaki wierszB, wierszB
wyswietl offset wierszB, wierszB
;--- ustawienie kursora
mov YX.X, 62
mov YX.Y, 29
INVOKE SetConsoleCursorPosition, hout, COORD PTR [YX] ; ustawienie pozycji kursora
;--- pętla
powtB:
;--- odczyt komunikatu zdarzenia
INVOKE FlushConsoleInputBuffer, hinp ; czyszczenie kolejki komunikatów
INVOKE ReadConsoleInputA, hinp, OFFSET zapis, 1, OFFSET nzdarz ; odczyt komunikatu zdarzenia
;--- sprawdzanie typu zdarzenia
cmp WORD PTR [zapis.EventType], KEY_EVENT
je @F
jmp powtB
@@:
jmp klawB
INVOKE SetConsoleCursorPosition, hout, COORD PTR [YX] ; ustawienie pozycji kursora
;--- komunikat od klawiatury ---
klawB:
cmp (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).bKeyDown, 0 ; porównanie z zerem
jne @F
jmp powtB ; jeśli nie ma naciśnięcia
@@:
mov AL, (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).ASCIIChar
mov symb, AL
cmp AL, 048h ; porównanie z kodem H
jne @F
jmp etykB
@@:

```

```

cmp AL, 068h ; porównanie z kodem h
jne @F
jmp etykB
@@:
INVOKE WriteConsoleOutputCharacterA,hout,OFFSET symb,1,COORD PTR [YX],OFFSET rout ; wypisywanie znaku
jmp powtB
etykB:
INVOKE FlushConsoleInputBuffer,hinp ;czyszczenie kolejki komunikatów
INVOKE ReadConsoleInputA,hinp,OFFSET zapis,1,OFFSET nzdarz ; odczyt komunikatu zdarzenia
cmp WORD PTR [zapis.EventType], KEY_EVENT
je @F
jmp etykB
@@:
jmp klaw2B
klaw2B:
cmp (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).bKeyDown, 0 ; porównanie z zerem
jne @F
jmp etykB ;jeśli nie ma naciśnięcia
@@:
mov AL,(KEY_EVENT_RECORD PTR [zapis.KeyEvent]).ASCIIChar
mov symb,AL
cmp AL, 049h ; porównanie z kodem l
jne @F
jmp konB
@@:
cmp AL, 069h ; porównanie z kodem i
jne @F
jmp konB
@@:
INVOKE WriteConsoleOutputCharacterA,hout,OFFSET symb,1,COORD PTR [YX],OFFSET rout ; wypisywanie znaku
jmp powtB
;--- zamknięcie konsoli
konB:
INVOKE FreeConsole
jmp start
;-----|
; Koniec zadania B |
;-----|

;-----|
; Zadanie C |
;-----|
zadanie_c:
;--- tworzenie konsoli ---
INVOKE FreeConsole
INVOKE AllocConsole
;--- nagłówek okna ---
INVOKE CharToOemA,OFFSET tytuł,OFFSET tytuł ; konwersja polskich znaków
INVOKE SetConsoleTitleA,OFFSET tytuł
;--- deskryptory buforów ---
INVOKE GetStdHandle, STD_OUTPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hout, EAX
INVOKE GetStdHandle, STD_INPUT_HANDLE ; wywołanie funkcji GetStdHandle
mov hinp, EAX
;--- obliczenie rozmiaru buforu ekranu
INVOKE GetLargestConsoleWindowSize,hout
mov YX, EAX
movzx EAX, YX.X
movzx EBX, YX.Y
mul EBX
mov rkom, EAX
;--- napelnienie komórek jednakowym atrybutem ---
mov YX.X, 0
mov YX.Y, 0
INVOKE FillConsoleOutputAttribute,hout,czerwony,rkom, \
COORD PTR [YX],OFFSET rfakt
;--- wyprowadzenie komunikatu informacyjnego ---
mov YX.X, 0
mov YX.Y, 29
INVOKE SetConsoleCursorPosition, hout, YX
INVOKE SetConsoleTextAttribute,hout,atryb1
plznaki wierszC, wierszC
wyswietl offset wierszC, wierszC
;--- ustawienie kursora
mov YX.X, 0
mov YX.Y, 1

```

```

INVOKE SetConsoleCursorPosition,hout,COORD PTR [YX] ;ustawienie pozycji kursora
;--- pętla
powtC:
;--- odczyt komunikatu zdarzenia
INVOKE FlushConsoleInputBuffer,hinp ;czyszczenie kolejki komunikatów
INVOKE ReadConsoleInputA,hinp,OFFSET zapis,1,OFFSET nzdarz ; odczyt komunikatu zdarzenia
;--- sprawdzanie typu zdarzenia
cmp WORD PTR [zapis.EventType], MOUSE_EVENT
je myszC
cmp WORD PTR [zapis.EventType], KEY_EVENT
je klawC
jmp powtC
;--- komunikat od myszy
myszC:
test (MOUSE_EVENT_RECORD PTR [zapis.MouseEvent]).dwButtonState, FROM_LEFT_1ST_BUTTON_PRESSED
jnz @F
jmp powtC ; na koniec programu
@@:
test (MOUSE_EVENT_RECORD PTR [zapis.MouseEvent]).dwControlKeyState, LEFT_ALT_PRESSED
jz @F
jmp konC
@@:
;--- jest naciśnięty prawy klawisz myszy
mov EAX, (MOUSE_EVENT_RECORD PTR [zapis.MouseEvent]).dwMousePosition ; współrzędne X,Y
mov COORD PTR [YX], EAX ; współrzędne X,Y
INVOKE SetConsoleCursorPosition,hout,COORD PTR [YX] ; ustawienie pozycji kursora
jmp powtC
;--- komunikat od klawiatury ---
klawC:
cmp (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).bKeyDown, 0 ; porównanie z zerem
jne @F
jmp powtC ;jeśli nie ma naciśnięcia
@@:
mov AL,(KEY_EVENT_RECORD PTR [zapis.KeyEvent]).ASCIIChar
mov symb,AL
cmp AL, 048h ; porównanie z kodem H
jne @F
jmp etykC
@@:
cmp AL, 068h ; porównanie z kodem h
jne @F
jmp etykC
@@:
INVOKE WriteConsoleOutputCharacterA,hout,OFFSET symb,1,COORD PTR [YX],OFFSET rout ; wypisywanie znaku
jmp powtC
etykC:
INVOKE FlushConsoleInputBuffer,hinp ;czyszczenie kolejki komunikatów
INVOKE ReadConsoleInputA,hinp,OFFSET zapis,1,OFFSET nzdarz ; odczyt komunikatu zdarzenia
cmp WORD PTR [zapis.EventType], MOUSE_EVENT
je myszC
cmp WORD PTR [zapis.EventType], KEY_EVENT
je @F
jmp etykC
@@:
jmp klaw2C
klaw2C:
cmp (KEY_EVENT_RECORD PTR [zapis.KeyEvent]).bKeyDown, 0 ; porównanie z zerem
jne @F
jmp etykC ;jeśli nie ma naciśnięcia
@@:
mov AL,(KEY_EVENT_RECORD PTR [zapis.KeyEvent]).ASCIIChar
mov symb,AL
cmp AL, 049h ; porównanie z kodem I
jne @F
jmp konC
@@:
cmp AL, 069h ; porównanie z kodem i
jne @F
jmp konC
@@:
INVOKE WriteConsoleOutputCharacterA,hout,OFFSET symb,1,COORD PTR [YX],OFFSET rout ; wypisywanie znaku
jmp powtC
;--- zamknięcie konsoli
konC:
INVOKE FreeConsole
jmp start

```

```

;-----|
; Koniec zadania C      |
;-----|
kon:
;--- zamknięcie konsoli
INVOKE FreeConsole
;---- wywołanie funkcji ExitProcess -----
invoke ExitProcess,0
;podprogramy
;-----|
; czyszczenie ekranu    |
;-----|
Cls proc
mov bl,1
@@:
    cmp bl, 50h
    je @F
    wyswietl nl,2
    add bl, 1
    jmp @B
@@:
mov bl,0
ret
Cls endp
;-----|
; Rysowanie menu programu      |
;-----|
Menu proc
;--- rysuje menu programu ----
wyswietl rogGL, 1 ; wyswietlenie lewego rogu tabelki
drukujZnak prosta, 118 ; wyswietla przyjetý znak 118 razy
wyswietl rogGP, 1 ; wyswietlanie prawego rogu tabelki
nowalinia nl, 2 ; makro
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 45
plznaki szkola1, szkola1
wyswietl offset szkola1, sz1
drukujZnak spacja, 45
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 49
plznaki szkola2, szkola2
wyswietl offset szkola2, sz2
drukujZnak spacja, 49
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 39
plznaki szkola3, szkola3
wyswietl offset szkola3, sz3
drukujZnak spacja, 38
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 46
plznaki szkola4, szkola4
wyswietl offset szkola4, sz4
drukujZnak spacja, 46
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 54
plznaki szkola5, szkola5

```


wyswietl offset szkola5, sz5
drukujZnak spacja, 53
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 34
plznaki szkola6, szkola6
wyswietl offset szkola6, sz6
drukujZnak spacja, 33
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1 ; wyświetlenie pionowej linii
drukujZnak spacja, 48 ; wyswietla przyjeta znak 118 razy
wyswietl offset menuTekst, menuR; tekst
drukujZnak spacja,48
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2 ; makro
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 43 ; ; wyświetlenie pionowej linii
plznaki zadanieA, zadanieA ; konwersja polskich znaków
wyswietl offset zadanieA, rozmA ; tekst
drukujZnak spacja, 43
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2 ; makro
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1 ; wyświetlenie pionowej linii
drukujZnak spacja, 43
plznaki zadanieB, buf ; konwersja polskich znaków
wyswietl offset buf, rozmB ; tekst
drukujZnak spacja, 43
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2 ; makro
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2 ; makro
wyswietl pion,1 ; wyświetlenie pionowej linii
drukujZnak spacja, 43
plznaki zadanieC, buf ; konwersja polskich znaków
wyswietl offset buf, rozmC ; tekst
drukujZnak spacja, 43
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2 ; makro
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 38 ; ; wyświetlenie pionowej linii
plznaki zakonczenie, zakonczenie ; konwersja polskich znaków
wyswietl offset zakonczenie, zakR ; tekst
drukujZnak spacja, 37
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjeta znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 49 ; ; wyświetlenie pionowej linii
plznaki wybrany, wybrany ; konwersja polskich znaków

```

wyswietl offset wybrany, wybrR; tekst
drukujZnak spacja, 47
wyswietl pion,1 ; wyświetlenie pionowej linii
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pion,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 45 ;; wyświetlenie pionowej linii
plznaki student, student ; konwersja polskich znaków
wyswietl offset student, stuR; tekst
drukujZnak spacja, 43
wyswietl pion,1
nowalinia nl, 2
wyswietl pionL,1
drukujZnak prosta, 118 ; wyswietla przyjetý znak 118 razy
wyswietl pionP,1
nowalinia nl, 2
wyswietl pion,1
drukujZnak spacja, 34 ;; wyświetlenie pionowej linii
plznaki profesor, profesor ; konwersja polskich znaków
wyswietl offset profesor, proR; tekst
drukujZnak spacja, 33
wyswietl pion,1
nowalinia nl, 2
wyswietl rogDL,1 ; wyswietlenie lewego dolnego rogu tabelki
drukujZnak prosta, 118 ; wyswietla przyjetý znak 118 razy
wyswietl rogDP,1; wyswietlenie prawego dolnego rogu tabelki
nowalinia nl,2
ret ; powrót do miejsca wywołania podprogramu
Menu endp
_text ends
end start
;-----|
; Koniec segmentu kodu |
;-----|

```

Testowanie

```

Warszawska Uczelnia Medyczna
im. Tadeusza Kořluka

Ćwiczenie z programowania niskopozomowego

Obsluga klawiatury i myszy
Wariant N 8

"Obsluga klawiszy H i I oraz Lewy Alt + lewy myszy"

.....: MENU :.....

Zadanie a) wciřnij klawisz [A/a]
Zadanie b) wciřnij klawisz [B/b]
Zadanie c) wciřnij klawisz [C/c]

Wciřnij klawisz [Q/q] aby wyjřć z programu.

Wybrales klawisz : [ ]

Wykonał: Grzegorz Makowski i53

Prowadzacy: Prof. dr hab. inż. Aleksandr Timofiejew

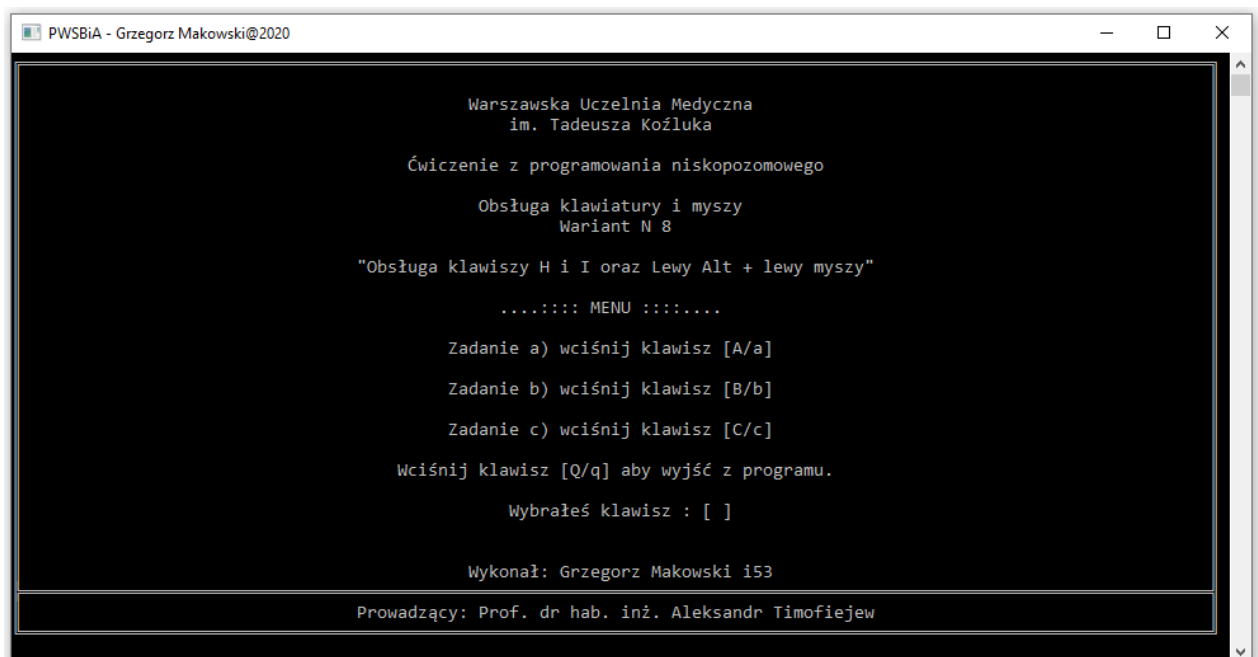
```

Windows 10 nie obsługuje zdarzenia FROM_LEFT_1ST_BUTTON_PRESSED dlatego zaimplementowano z zadania b dodatkową opcję wyjścia klawiszami H i I



```
PWSBiA - Grzegorz Makowski@2020

Zadanie można zakończyć klawiszami Lewy Alt + Lewy przycisk myszy lub klawiszami H, I lub h, i
```



```
PWSBiA - Grzegorz Makowski@2020

Warszawska Uczelnia Medyczna
im. Tadeusza Koźłuka

Ćwiczenie z programowania niskopozomowego

Obsługa klawiatury i myszy
Wariant N 8

"Obsługa klawiszy H i I oraz Lewy Alt + lewy myszy"

.....::: MENU :::.....

Zadanie a) wciśnij klawisz [A/a]
Zadanie b) wciśnij klawisz [B/b]
Zadanie c) wciśnij klawisz [C/c]

Wciśnij klawisz [Q/q] aby wyjść z programu.

Wybrałeś klawisz : [ ]

Wykonał: Grzegorz Makowski i53

Prowadzący: Prof. dr hab. inż. Aleksandr Timofiejew
```