

# ~~~~~Zaoczni + 1 KOLOS~~~~~

**Z1. Użycie języka assemblera do programowania można rekomendować, gdy program jest:**

- a) ogólnego przeznaczenia na jedną platformę komputerową
- b) ogólnego przeznaczenia na wiele platform komputerowych
- c) **systemem wbudowanym na jedno urządzenie**
- d) **sterownikiem urządzenia komputera**
- e) **procedurą obsługi przerwania lub wyjątku**
- f) systemem operacyjnym komputera

**Z2. Wskaż skutki użycia dyrektywy REPEAT (w języku assemblera) w następującej konstrukcji:**

REPEAT wyrażenie

instrukcja

ENDM

- a) Wykonanie "instrukcja" tyle razy ile wskazuje wyrażenie
- b) **utworzenie kodu "instrukcja" tyle razy ile wskazuje wyrażenie**
- c) wykonywanie "instrukcja" tak długo, aż wyrażenie uzyska wartość true
- d) **wydłużenie tekstu programu źródłowego w stosunku do wersji program bez użycia dyrektywy REPEAT**
- e) skrócenie czasu wykonania programu w stosunku do wersji program bez użycia dyrektywy REPEAT
- f) utworzenie kodu "instrukcja" tak długo aż wyrażenie uzyska wartość TRUE

**Z3. Makrodefinicja ma następującą postać:**

```
MA          MACRO X1,X2
              MOV AX,X2
              MOV X1,AX          ENDM
```

Jak będzie wyglądać rozwinięcie makrowywołania: MA D1, D2

- a) **MOV X2,D2**  
MOV AX,X2  
MOV X1,AX  
MOV D1,X1
- b) **MACRO D1,D2**  
MOV AX,X2  
MOV X1,AX  
ENDM
- c) **MOV AX,X2**  
MOV X1,AX
- d) **MOV AX,D1**  
MOV D2,AX
- e) **MOV AX,D2**  
MOV D1,AX  
ENDM
- f) **MOV AX,D2**  
**MOV D1,AX**

**Z4. Wskaż zdanie prawdziwe dotyczące dyrektyw assemblera warunkowej:**

- A) Użycie dyrektyw assemblera warunkowej umożliwia wykonanie bloków programu w zależności od warunków sprawdzanych przez program.
- B) **Wybór bloków programu źródłowego, które zostaną przetłumaczone na ML, zależy od kombinacji warunków określonych dyrektywami assemblera warunkowej**
- C) Blok programu źródłowego objęty dyrektywami assemblera warunkowej będzie tłumaczony w zależności od warunków sprawdzanych przez program
- D) Assembler warunkowa wykorzystuje instrukcje warunkowe ML w celu umożliwienia wariantowości programu
- E) **Wybór bloków programu źródłowego, które zostaną przetłumaczone na ML następuje w czasie assemblera programu źródłowego**
- F) Jeżeli w dyrektywach assemblera warunkowej zastąpić stałe zmiennymi, to bloki programu wyznaczone tymi dyrektywami będą wykonywane warunkowo, w zależności od stanów warunków wyznaczonych w czasie działania programu.

**Z5. Wskaż zdania prawdziwe dotyczące makroinstrukcji.**

- A) Makroinstrukcja jest nazwanym ciągiem instrukcji ML, który zostanie wstawiony przez asembler do programu źródłowego, ilekroć asembler napotka nazwę tej makroinstrukcji
- B) Użycie Makroinstrukcji daje identyczny skutek jak użycie procedury //dopytać ZBYSZKA jak rozumie SKUTEK(nie trzeba dopytywać napewno to adp nie pasuje bo przecież w macro nie ma angażowania stosu)<-(ale rezultat będzie taki sam)
- C) Użycie Makroinstrukcji jest mniej wygodne niż użycie procedury, gdyż w przypadku Makroinstrukcji nie można przekazywać parametrów
- D) Makroinstrukcje służą do utworzenia modułów
- E) Używanie makroinstrukcji może przyspieszyć działanie programu, w stosunku do wersji używającej procedur w miejsce Makroinstrukcji
- F) Używanie Makroinstrukcji może wydłużyć kod źródłowy programu, w stosunku do wersji używającej procedur w miejsce Makroinstrukcji

**Z6. Zaznacz zdania prawdziwe dotyczące danej utworzonej w języku asemblera dyrektywą generacji danej użytą w konstrukcji:**

**SYMBOL: DB <wyrażenie>**

- A) SYMBOL uzyska wartość <wyrażenie>
- B) Obraz komórki pamięci, której lokalizacja oznaczona została przez SYMBOL, uzyska wartość <wyrażenie>
- C) program wpisze do pamięci wartości <wyrażenia>
- D) Wartością SYMBOL jest adres komórki w pamięci, gdzie zostanie umieszczona wartość <wyrażenie>
- E) Zmiana w czasie wykonania programu wartości zmiennych występujących w <wyrażenie> zmieni wartość danej generowanej dyrektywą
- F) <wyrażenie> musi być wyliczane w czasie asemblacji programu

**Z7. Rozgałęzienie warunkowe od wyniku porównania  $a \geq b$  (a,b bez znaku) powinno wykorzystać flagę(i) w następujący sposób:**

- A) Skok, gdy flaga przepełnienia ustawiona
- B) Skok, gdy flaga przeniesienia ustawiona
- C) Skok, gdy flaga zera ustawiona
- D) Skok, gdy flaga przepełnienia wyzerowana i flaga zera ustawiona
- E) Skok, gdy flaga przeniesienia wyzerowana
- F) Skok, gdy flaga znaku wyzerowania lub flaga zera ustawiona

**Z8. Kiedy program wykonujący obliczanie arytmetyczne należy zakończyć rozgałęzieniem do obsługi błędu obliczania wykonywanym, gdy flaga przepełnienia jest ustawiona?:**

- A) Gdy obliczanie polega na sumowaniu liczb bez znaku
- B) Jeżeli  $a < b$ , a obliczanie polega na  $a - b$
- C) Gdy obliczanie polega na odejmowaniu liczb ze znakiem
- D) Gdy obliczanie polega na sumowaniu liczb ze znakiem
- E) Gdy obliczanie polega na odejmowaniu liczb bez znaku
- F) Jeżeli wynik operacji przekroczy zakres określony liczbą bitów reprezentacji liczby

**Z9. Jeżeli, w odniesieniu do parametru przekazywanego pomiędzy programem a procedurą: "procedura nie modyfikuje zmiennej odpowiadającej temu parametrowi", to:**

- A) Taki parametr określamy jako "wejściowo-wyjściowy"
- B) Taki parametr nie może być przekazany poprzez zmienną
- C) Taki parametr określamy jako "wyjściowy"
- D) Taki parametr określamy jako "stałą programu"
- E) Taki parametr nie może być przekazany poprzez odwołanie (referencję)
- F) Taki parametr określamy jako "wejściowy"

**Z10. Wskaż zdania poprawne dotyczące konstrukcji pętli WHILE**

- A) Konstrukcja zawiera: skok warunkowy poza pętlę; na końcu pętli sprawdzanie warunku i skok bezwarunkowy na początek
- B) Konstrukcja zawiera: sprawdzanie warunku, skok warunkowy poza pętlę; na końcu pętli skok bezwarunkowy na początek
- C) Konstrukcja zawiera: sprawdzanie warunku, skok bezwarunkowy poza pętlę; na końcu pętli skok warunkowy na początek
- D) Instrukcja skoku warunkowego musi być poprzedzona etykietą
- E) Instrukcja sprawdzania warunku musi być poprzedzona etykietą
- F) Pierwsza instrukcja po pętli nie wymaga etykiety

**Z11. Które ze zdań odnoszących się do przekazywania parametru w "kodzie programu" nie są prawdziwe:**

- A) Możliwe jest przekazywanie zmiennych o dowolnie zmiennej długości
- B) Jeżeli kod programu umieszczony jest w segmencie bez prawa zapisu, wówczas to samo wywołanie procedury nie może się odbyć z każdorazowo inną wartością parametru
- C) Parametr umieszcza się bezpośrednio po instrukcji wywołania procedury
- D) Nie wolno w procedurze modyfikować zapisanego na stosie adresu powrotu**
- E) Parametr można umieścić bezpośrednio po dowolnej instrukcji programu
- F) W każdej architekturze procesora parametry przekazywane tą drogą są tylko stałymi**

**Z12. Zaznacz prawdziwe zdania odnoszące się do wykonania działania A+C, gdzie Ah jest starszą a Al młodszą częścią liczby A; Ah, Al, C mają długość 8 bitów; procesor wykonuje operacje arytmetyczne wyłącznie na danych 16 bitowych**

- A) A+C oblicza się pojedynczą instrukcją procesora
- B) Należy dodać C do Al, a następnie do Ah
- C) Należy rozszerzyć C do formatu A i dodać pojedynczą instrukcją**
- D) Należy utworzyć 16 bitowy odpowiednik C, w którym starszy bajt zawiera same zera
- E) Należy utworzyć 16 bitowy odpowiednik C, w którym starszy bajt zawiera same jedynki
- F) Należy dodać C do Al, a następnie dodać 0 do Ah z uwzględnieniem przeniesienia z dodawania C do Al

**Z13. Opisz w krokach jak zaimplementować pętlę WHILE w procesorze, w którym wszystkie rozkazy wykonywane są warunkowo (ARM)**

0. inicjalizacja zmiennej sterującej (poprzez np zapisanie do rejestru ogólnego przeznaczenia jakiejś dodatniej wartości)
1. napotkanie etykiety (np petla\_while)
2. wykonywanie instrukcji
3. inkrementacja licznika pętli
4. porównanie zmiennej z rejestru OP ze zmienną sterującą
5. Skok poza pętlę, jeśli wartości równe, w przeciwnym razie 6
6. skok do etykiety petla\_while

; Display integers 1 – 10:

```
mov eax,0
.WHILE eax < 10
inc eax
call WriteDec
call CrLf
.ENDW
```

```
mov eax,0
@WHILE:
//warunek
jae @C0001 ; fałsz? Opuść pętlę
//instrukcja
jnb @WHILE
@C0001:
```

**Z13. Opisz w krokach, jak zaimplementować przesunięcie o 1 pozycję w lewo słowa 16-to bitowego w procesorze w którym występują wyłącznie 8-bitowe przesunięcia**

```
shl al, 1
rcl ah, 1
```

## ~~~~~NASZ II KOŁOS~~~~~

**Z1 Konsolidacja polega na:**

- a) **Scałeniu różnych plików źródłowych w jeden plik wynikowy.**
- b) Załadowaniu różnych modułów we wspólny obszar pamięci.
- c) Przypisaniu wartości symbolom zdefiniowanym poza bieżącym modułem.
- d) Zapisaniu w pliku wynikowym tablicy symboli eksportowanych.

**Z2. Wskaż zdanie NIEprawdziwe dotyczące deklarowania procedury dyrektywą „PROC”**

- a) Użycie „PROC” przy etykietce wejściowej procedury służy wskazaniu, od którego miejsca w programie rozciąga się lokalność zmiennych zadeklarowanych w procedurze.
- b) **użycie „PROC” z następującą po niej listą nazw parametrów spowoduje, że parametry z listy zostaną zmiennymi lokalnymi(wewn.) i zostanie utworzona odpowiednia ramka na stosie**
- c) użycie „PROC” z następującą po niej listą parametrów umożliwia zadeklarowanie parametrów przekazywanych między programem wywołującym, a procedurą i określenie ich typów
- d) dyrektywa „PROC” powinna być poprzedzona etykietą (nazwą) procedury

**Z3.a Które ze zdań odnoszących się do przekazywania parametru w „kodzie programu” NIE jest prawdziwe:**

- a) Możliwe jest przekazywanie zmiennych o zmiennej długości
- b) **Jeżeli kod programu umieszczony jest w segmencie bez prawa zapisu, wówczas to samo wywołanie procedury nie może się odbyć z każdorazowo inną wartością parametru**
- c) Parametr umieszcza się bezpośrednio po instrukcji wywołania procedury
- d) **nie wolno w procedurze modyfikować adresu powrotu, który został zapisany na stosie w wyniku wywołania procedury**

**Z4. Wskaż prawdziwe zdanie dotyczące deklarowania ciała procedury dyrektywą taką jak „ENDP”**

- a) Musi wystąpić na końcu każdego rozgałęzienia procedury
- b) zastępuje instrukcję RET
- c) Umożliwia assemblerowi określić zakres lokalności wewnętrznych zmiennych procedury
- d) **Stosowana jednocześnie z dyrektywą PROC umożliwia assemblerowi określić zakres lokalności wewn. Zmiennych procedury**

**Z5. Relokacji wymagają:**

- a) wszystkie instrukcje maszynowe
- b) instrukcje z adresowaniem natychmiastowym
- c) instrukcje z adresowaniem względnym
- d) **wszystkie instrukcje z wyjątkiem instrukcji określonych w pkt B i C powyżej**

**Z6. Relokacja kodu programu podczas ładowania programu może polegać na:**

- a) zastąpieniu dotychczasowych kodów rozkazów nowymi kodami rozkazów wynikających z bieżącego rozmieszczenia programu w pamięci
- b) dodaniu adresu położenia początku programu do adresów odwołań polach argumentów tych rozkazów, które zostaną rozmieszczone w innych lokalizacjach, niż lokalizacje ustalone podczas asemlacji programi
- c) **Dodaniu adresu położenia początku programu do adresów odwołań w polach argumentów tych rozkazów, które odwołują się do lokalizacji zmienionych w stosunku do położenia ustalonego podczas asemlacji programu**
- d) nadanie nowych wartości adresów etykietom występującym w programie, poprzez dodanie adresu położenia początku programu

**Z7. Wskaż zdanie prawdziwe dotyczące traktowania wyrażeń dla potrzeb relokacji:**

- a) różnica wyrażania względnego i bezwzględnego ma wartość bezwzględną (wyrażenie nierelokowalne)
- b) różnica wyrażeń bezwzględnych ma wartość bezwzględną (wyrażenie nierelokowalne)
- c) różnica wyrażeń względnych ma wartość względną (wyrażenie relokowalne)
- d) **różnica wyrażeń względnych ma wartość bezwzględną (wyrażenie nierelokowalne)**

**Z8. Jeżeli parametry są przekazywane do procedury poprzez stos, wówczas prawdziwe jest, że:**

- a) **W celu przekazania parametrów należy zarezerwować na stosie „ramkę” na odpowiednią liczbę bajtów i wywołać procedurę**
- b) **należy opróżnić stos z parametrów bądź w procedurze, bądź na wyjściu z procedury w programie głównym**
- c) w ciele procedury pierwszą operacją powinno być odczytanie pierwszego parametru ze stosu instrukcją POP (pobierz ze stosu)
- d) w ciele procedury możliwe jest odczytywanie parametrów bez usuwania ich ze stosu

**Z9. Jeżeli w programie znajduje się dyrektywa taka, jak INVOKE: z nazwą procedury i następującą po niej listą parametrów, to:**

- a) makroassembler umieści w kodzie programu instrukcje zapisania parametrów na stos i wywołania procedury
- b) makroassembler umieści w kodzie programu instrukcję wywołania procedury, a w dalszej kolejności instrukcja tworzące ramkę na stosie dla parametrów lokalnych
- c) makroassembler nie utworzy żadnych instrukcji poza instrukcją „wywołania procedury”, natomiast sprawdzi zgodność typów parametrów z zadeklarowanymi dyrektywą prototypu procedury („PROTO”) i w przypadku niezgodności zasygnalizuje błąd
- d) makroassembler zastąpi instrukcje odwołujące się do symboli przekazywanych parametrów instrukcjami odwołującymi się do lokalizacji parametrów na stosie

**Z10. Jeżeli procedura przekazuje wyniki na stosie, to:**

- a) żadne z poniższych
- b) wykonuje instrukcję RET, a następnie zapisuje na stos wyniki tak, aby były dostępne w programie wywołującym
- c) zapisuje wyniki na stos, następnie wskaźnik do lokalizacji wyników na stosie wpisuje do wybranego rejestru, ustawia wskaźnik stosu tak, aby wskazywał adres powrotu wykonuje instrukcję RET
- d) zapisuje na stos wyniki, a następnie wykonuje instrukcję RET

**Z11. Które ze zdań odnoszących się do implementacji zmiennych lokalnych (procesory Intel 286+) jest prawdziwe:**

- a) użycie dyrektywy „LOCAL” z listą nazw parametrów jest równoważne użyciu instrukcji ENTER z parametrem całkowitym
- b) użycie dyrektywy „LOCAL” z listą nazw parametrów jest równoważne użyciu instrukcji LEAVE
- c) użycie dyrektywy „LOCAL” z listą nazw parametrów jest równoważne użyciu pary instrukcji :ENTER (po wejściu do procedury) i LEAVE (przed wyjściem z procedury)
- d) działaniu dyrektywy „LOCAL” nie można przypisać żadnego równoważnego kodu ASM

**Z12. Wskaż właściwą kolejność działań wykonywanych, gdy biblioteka systemowa jest ładowana dynamicznie przez system operacyjny**

**X- procedura biblioteczna, Y-loader dynamiczny, Z- wywołanie systemowe, W-obługa wywołania, Q-program główny**

- a) X-Y-Z-W-Q
- b) Q-Y-W-X-Q
- c) Q-X-Q
- d) Q-Z-W-Y-X-Y-Z-Q

**Z13. Które ze zdań odnoszących się do dynamicznego tworzenia zmiennych lokalnych procedury, NIE jest prawdziwe:**

- a) tą samą procedurę, z dynamicznie utworzonymi zmiennymi lokalnymi mogą wywoływać różne równolegle wykonywane programy
- b) aby dynamicznie utworzyć zmienne procedury, należy zarezerwować odpowiednią ilość miejsca na stosie i wywołać procedurę
- c) procedurę, z tak utworzonymi zmiennymi lokalnymi, można wywoływać rekurencyjnie
- d) W czasie działania procedury nadrzędnej dynamicznie zmienne lokalne nie mają lokalizacji w pamięci, dlatego nie można się do nich odwołać z proced

**Z14. Co zawiera Program Linking Table (PLT) w pliku wynikowym w formacie ELF?**

- a) kody skoków z adresowaniem pośrednim pamięciowym do dynamicznie zmiennych adresów
- b) adresy ustalone podczas konsolidacji dynamicznej
- c) tablice symboli importowanych
- d) tablicę symboli eksportowanych

**Z15. Jaką postać powinny mieć moduły programu, aby mogły zostać poddane konsolidacji statycznej?**

- a) programu źródłowego (plik tekstowy)
- b) relokowalnego pliku wynikowego z tablicą symboli importowanych i eksportowanych
- c) pliku (wynikowego) ładowalnego
- d) relokowalnego pliku wynikowego z tablicą symboli importowanych

**Z16. Jaką informacją musi dysponować assembler, żeby przetłumaczyć do postaci konsolidowanej moduł zawierający odwołania do symboli importowanych**

- a) wartościami symboli
- b) adresami odwołań do symboli
- c) Formatem symboli
- d) identyfikacją modułu, w którym symbol zdefiniowano

**Z17. Wskaż zdanie prawdziwe dotyczące konsolidacji programów**

- a) importowany punkt wejścia jest relokowalny
- b) eksportowany punkt wejścia jest relokowalny
- c) eksportowany symbol jest relokowalny

- d) importowany symbol jest relokowalny

**Z18. Rekordy modyfikacji w pliku konsolidowanym:**

- a) identyfikację sposobu relokacji symboli
- b) identyfikację operacji, którą ma wykonać z użyciem symbolu eksportowanego
- c) instrukcję w której należy umieścić symbol importowany
- d) identyfikację operacji, którą ma wykonać z użyciem symbolu importowanego