1. Konwersja znaku w formacie UTF-8 na znak w formacie UTF-16 dla punktów kodowych z przedziału od 10000h do 10FFFFh (1F514 → 3D D8 14 DD)
2. Konwersja znaku w formacie UTF-16 na znak w formacie UTF-8 dla punktów kodowych z przedziału od 10000h do 10FFFFh (1F514 → 3D D8 14 DD)
3. Podprogram, który podzieli liczbę w kodzie U2 z rejestru AL przez -2. Resztę z dzielenia wpisać do znacznika CF, a iloraz do rejestru AL. Dzielenie wykonaj tylko za pomocą rozkazu SAR
4. W czterech kolejnych bajtach pamięci począwszy od adresu w ESI znajduje się liczba 32 bitowa bez znaku kodowane według big endian, zapisz tą liczbę to rejestru EDI w formacie little endian bez wykorzystania rozkazu BSWAP
5. Napisz program, który zamieni liczbę w kodzie U2 zapisaną w rejestrze AL na liczbę w kodzie minus dwójkowym. Przy dzieleniu przez -2 wykorzystaj program nr 3
6. Napisz program, który obliczy liczbę bitów o wartości 0 zawartych w rejestrze EAX. Wynik obliczenia wpisać do rejestru CL. Wykorzystaj → ADC CL, 0
7. Jaka wartość zostanie wpisana do rejestru EAX po wykonaniu poniższych rozkazów

```
pqr     dw      65, 129, 257
- - - -
mov EBX, offset pqr
mov EDI, 1
mov EAX, [EDI][EBX]
```

8. Jaka wartość zostanie wpisana do rejestru EAX po wykonaniu poniższych rozkazów

```
prr     dw      65, 129, 257
- - - -
mov EBX, offset prr
mov ESI, 2
mov EAX, [EBX + ESI - 1]
```

9. Napisz program, który zarezerwuje 64 bajty na zmienną przeznaczoną do reprezentacji łańcucha znaków UTF-16 i odwoła się do 23 znaku tego łańcucha. Przyjąć że znaki są z przedziału od 0h do FFFFh
10. Rejestr EAX wynosi 156587h. Podaj zawartość znaczników OF, ZF i CF po wykonaniu fragmentu programu

```
shl EAX, 0Bh
sub EAX, 4000 0000h
```

11. Jaka wartość zostanie wpisana do rejestru EAX po wykonaniu poniższych rozkazów

```
czw     dd      0, 1, 2
pt      dw      3, 4, 5, 6
sb      dw      7, 8, 9, 10, 11, 12, 13
- - - -
mov EDI, offset pt - offset czw
mov EAX, dword ptr sb[edi+2]
```

12. Napisz program, który wczyta 3 bit z pamięci o adresie podanym w ESI i bitu o numerze podanym w CL i wpisze je na 3 najmłodsze bity rejestru AL
13. Podaj zawartość znaczników OF, ZF i CF po wykonaniu fragmentu programu

```
mov EAX, 0B193h
add AH, AL
```

14. Ile bajtów zarezerwuje assembler na zmienne opisane poniżej

```
        db      0Ah, dup(0)
        dw      'A' - 'Z'
pon     dq      2.5
```

15. Podaj równoważny fragment kodu, tak by nie skorzystać z rozkazów grupy SET

```
and EAX, 0
setae CL
```

16. Zakładając, że ESP wskazuje na początek łańcucha znaków ASCII oraz to, że jest definicja zmiennej "znak db ?" podaj fragment kodu, który umieści w zmiennej znak wartość elementu łańcucha o indeksie zawartym w rejestrze DL
17. Napisz program do szukania największego wspólnego dzielnika dla liczb 32 bitowych w ESI i EDI według algorytmu

```
while (y != 0) {
        x = x % y;
        swap(x,y)
}
```

18. Napisz program, który pobiera 9 bitów najmłodszych z rejestru BX i zapisuje je w pamięci od bajtu o adresie podanym w ESI i bitu podanym w rejestrze DL. Pozostałe bity w bieżącym i sąsiednich bajtach nie mogą być zmienione.
19. Napisz program, który pobiera 5 bitów z pamięci od bajtu o adresie podanym w ESI i bitu podanym w rejestrze CL i wpisuje je do do AL w postaci 000x xxxx
20. W programie zdefiniowano zmienną "obszar dw 2 dup (?). Wpisz do tej tablicy początkowy adres obszaru pamięci, gdzie ona się znajduje.
21. Napisz program, który zamieni skompresowany tekst małych znaków ASCII i spacji na normalny. Znaki są 5 bitowe i zajmują wartości od 0 do 27, przy czym 0 to koniec ciągu, a 27 to spacja. Znaki ASCII zmniejszono o 60h. Adres skompresowanego tekstu to ESI, a wynikowego to EDI. Tekst normalny ma kończyć się bajtem zerowym. Wykorzystaj program numer 20.
22. Napisz program, który sprawdzi, czy liczba w EBX jest nieparzysta. Jeśli tak to wpisz do CF jedynkę, jeśli nie to wpisz tam zero (rozkaz STC i CLC)
23. Określ komunikat jaki zostanie wyświetlony

```
tekst   dw 'ar', 'ch', 'it', 'ek', 'tu', 'ra', 0
— — — — — — — —
push 0
push OFFSET tekst               ; tytuł
lea eax, dword ptr [tekst+6]
push eax                        ; treść
push 0
call _MessageBoxA@16
```

24. Zdefiniowano zmienną "obszar dw 2 dup(?)". Wpisz do tej tablicy adres aktualnie wykonywanej instrukcji

DODATKOWO ZADANIA PRZYKŁADOWE Z MOODLE