

## **Опрацювання рядка символів засобами асемблера мікропроцесорів x86. Робота з файлами**

**МЕТА РОБОТИ:** освоїти команди асемблера для роботи з рядками символів; опанувати функції Win32 для роботи з файлами; розвинути навички складання програми для опрацювання рядка символів та програми для створення, записування і читання текстового файла; відтранслювати і виконати в режимі відлагодження програми, складені відповідно до свого індивідуального завдання.

### **1. Теоретична частина**

#### **I. Команди оброблення рядкових примітивів**

В системі команд процесорів Intel передбачено п'ять груп команд для оброблення масивів байтів, слів та подвійних слів. Незважаючи на те, що всі вони називаються рядковими примітивами, область їх використання не обмежується тільки масивами рядків. З огляду на це доцільніше використовувати їх іншу назву – ланцюжкові команди.

Для адресації пам'яті в цих командах використовуються регістри ESI та EDI. Особливість цих команд полягає в тому, що обидва операнди розташовані в пам'яті. При обробленні рядкових примітивів ці команди можуть автоматично повторюватися, що робить їх застосування особливо зручним для роботи з довгими рядками та масивами.

При роботі програми в захищеному режимі адресація пам'яті в командах оброблення рядкових примітивів може здійснюватися через регістри ESI або EDI. При цьому зміщення, що міститься в регістрі ESI, відраховується відносно сегмента, чий дескриптор вказаний в регістрі DS, а зміщення, вказане в регістрі EDI, відраховується відносно сегмента, чий дескриптор вказаний в регістрі ES. При використанні лінійної моделі пам'яті в сегментних регістрах DS та ES міститься одне і те ж значення, яке в програмі неможна змінювати.

**Використання префікса повторення.** Самі по собі команди оброблення рядкових примітивів виконують тільки одну операцію над байтом, словом або подвійним словом пам'яті. Однак, якщо перед ними вказати префікс повторення, виконання команди буде повторено стільки разів, скільки вказано в регістрі ECX. Тобто з допомогою префікса можна виконати оброблення цілого масиву за допомогою всього однієї команди. Існує кілька типів префіксів повторення:

**REP** - Повторювати команду, поки ECX > 0;

**REPZ, REPE** - Повторювати команду, поки ECX > 0 і прапорець нуля установлений (ZF = 1);

**REPNZ, REPNE** - Повторювати команду, поки ECX > 0 і прапорець нуля скинутий (ZF = 0).

**Прапорець напрямку DF.** Стан цього прапорця впливає на те, який напрямок переміщення по рядку і як в процесі виконання команд оброблення рядкових примітивів змінюються значення регістрів ESI та EDI. Якщо

прапорець DF скинутий (напрямок - прямий), вони збільшуються на розмір оброблюваного операнда (1, 2 або 4 байти), а якщо встановлений (напрямок - зворотний), то відповідно зменшуються.

Значення прапорця напрямку DF можна явно задати за допомогою команд CLD та STD:

CLD	; Скидає прапорець напрямку DF (напрямок – прямий)
STD	; Встановлює прапорець напрямку DF (напрямок - зворотний)

### 1. Команди MOVSB, MOVSW та MOVSD

Ці команди дозволяють скопіювати дані з однієї ділянки пам'яті, адреса якого вказана в регістрі ESI, в іншу ділянку пам'яті, адреса якого вказана в регістрі EDI. При цьому, залежно від стану прапорця напрямку, значення в регістрах ESI та EDI або збільшується, або зменшується.

**MOVSB** – копіює послідовність байтів.

**MOVSW** – копіює послідовність слів.

**MOVSD** – копіює послідовність подвійних слів.

З командами MOVSB, MOVSW і MOVSD може використовуватися префікс повторення. При цьому значення регістрів ESI та EDI будуть автоматично змінюватися залежно від стану прапорця напрямку і типу команди.

**Приклад:** копіювання масиву подвійних слів. Потрібно скопіювати масив, що складається з двадцяти подвійних слів зі змінної source в змінну target.

```
.data
source DWORD 20 DUP (0FFFFFFFFh)
len_source EQU $- source
target DWORD 20 DUP (?)
.code
cld ; Скинемо прапорець DF і встановимо прямий напрямок
mov ECX, len_source ; Задамо значення лічильника
mov ESI, OFFSET source ; Задамо адресу джерела даних
mov EDI, OFFSET target ; Задамо адресу одержувача даних
rep movsd ; Копіюємо 20 подвійних слів
```

### 2. Команди CMPSB, CMPSW та CMPSD

Ці команди дозволяють порівняти дані з однієї ділянки пам'яті, адреса якої вказана в регістрі ESI, з іншою ділянкою пам'яті, адреса якої вказана в регістрі EDI.

**CMPSB** – порівнює послідовність байтів.

**CMPSW** – порівнює послідовність слів.

**CMPSD** – порівнює послідовність подвійних слів.

З командами CMPSB, CMPSW і CMPSD може використовуватися префікс повторення. При цьому значення регістрів ESI та EDI будуть автоматично змінюватися залежно від стану прапорця напрямку і типу команди.

### 3. Команди SCASB, SCASW і SCASD

Ці команди порівнюють значення, що міститься в регістрах AL/AX/EAX з байтом, словом або подвійним словом, які адресуються через регістр EDI.

Дана група команд зазвичай використовується для пошуку деякого значення в довгому рядку або масиві. Якщо перед командою SCAS помістити префікс REPE (або REP), рядок або масив скануватиметься до тих пір, поки значення в регістрі ECX не стане рівним нулю, або поки не буде знайдено значення в рядку або масиві, відмінне від того, що міститься в регістрі AL/AX/EAX (тобто поки не буде скинутий прапорець нуля ZF). При використанні префікса REPNE, рядок або масив скануватиметься до тих пір, поки значення в регістрі ECX не стане рівним нулю, або поки не буде знайдено значення в рядку або масиві, що збігається з тим, яке міститься в регістрі AL/AX/EAX (тобто поки не буде встановлено прапорець нуля ZF).

Пошук символів в рядку. У наведеному нижче фрагменті коду виконується пошук символу "F" в рядку alpha. При знаходженні даного символу, в регістрі EDI буде міститися його адреса плюс одиниця. Якщо ж шуканого символу немає в заданому рядку, то робота програми завершується в результаті переходу по команді JNZ.

```
.data
alpha BYTE "ABCDEFGH", 0
len_alpha EQU $-alpha
.code
mov EDI, OFFSET alpha      ; Завантажимо в EDI адресу рядка alpha
mov AL, 'F'                ; Завантажимо в AL ASCII-код символу "F"
mov ECX, len_alpha         ; Завантажимо в ECX довжину рядка alpha
cld                        ; Напрямок порівняння - прямий
repne scasb; Скануємо рядок поки не знайдемо символ "F"
jnz quit; Якщо не знайшли, завершимо роботу
dec EDI                    ; Символ знайдений, його адреса в EDI
```

У цьому прикладі після команди repne scasb є команда умовного переходу jnz, яка спрацьовує в разі, коли символ "F" в заданому рядку знайдений не буде (тобто коли робота команди repne scasb завершиться за умовою ECX = 0, а не ZF = 1).

#### 4. Команди STOSB, STOSW і STOSD

Ця група команд дозволяє зберегти вміст регістра AL/AX/EAX в пам'яті, яка адресується через регістр EDI. При виконанні команди STOS вміст регістра EDI змінюється відповідно до значення прапорця напрямку DF і типу використовуваного в команді операнда. При використанні спільно з префіксом REP, за допомогою команди STOS можна записати одне і те ж значення в усі елементи масиву або рядки. До прикладу, у наведеному нижче фрагменті коду виконується ініціалізація рядка string1 значенням 0FFh:

```
.data
Count = 100
string1 BYTE Count DUP (?)
.code
mov AL, 0FFh; записується значення
mov EDI, OFFSET string1 ; Завантажимо в EDI адресу рядка
```

```

mov ECX, Count ; Завантажимо в ECX довжину рядка
cld            ; Напрямок переміщення – прямий
rep stosb     ; Заповнимо рядок вмістом AL

```

## 5. Команди LODSB, LODSW і LODSD

Ця група команд дозволяє завантажити в регістр AL/AX/EAX вміст байта, слова або подвійного слова пам'яті, що адресується через регістр ESI. При виконанні команди LODS вміст регістра ESI змінюється відповідно до значення прапорця напрямку DF і типом операнда, який використовується в команді. Префікс REP практично ніколи не використовується з командою LODS, оскільки при цьому буде губитися попереднє значення, завантажене в акумулятор. Таким чином, ця команда використовується для завантаження одного значення в акумулятор.

Множення елементів масиву. У наведеній нижче програмі кожен елемент масиву подвійних слів array множиться на постійне значення. Для завантаження в регістр EAX поточного елемента масиву використовується команда LODSD, а для збереження - STOSD.

```

TITLE Множення елементів масиву (Mult.asm)
; У цій програмі кожен елемент масиву подвійних слів
; множиться на постійне значення.
.data
array DWORD 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
len_array EQU $-array
multiplier DWORD 10
.code
main PROC
cld ; Напрямок - прямий
mov ESI, OFFSET array ; Завантажимо адресу масиву
mov EDI, ESI ; в регістри ESI та EDI
mov ECX, len_array ; Завантажимо довжину масиву
LI:
Lodsd ; Завантажимо поточний елемент масиву
; в регістр EAX (його адреса в регістрі ESI)
mul multiplier ; Помножимо його на константу
stosd ; Запишемо EAX в поточний елемент
; масиву (його адреса в EDI)

loop LI
exit
main ENDP
END main

```

## II. Функції Win32 для роботи з файлами

### Створення файла

Приклад: програма WriteFile.asm

Нижче наведена програма WriteFile.asm, в якій створюється новий файл, і в нього записується деякий текст. При створенні файлу використовується опція

CREATE\_ALWAYS, тому якщо файл з таким ім'ям вже існує, його вміст стирається.

```
TITLE Використання функції WriteFile (WriteFile.asm)
INCLUDE Irvine32.inc

.data
buffer BYTE "Цей текст буде записаний в файл.", 0dh, 0ah
bufSize = ($ -buffer)
errMsg BYTE "Помилка при створенні файлу.", 0dh, 0ah, 0
filename BYTE "output.txt", 0
fileHandle DWORD ? ; Дескриптор файлу для записування
bytesWritten DWORD ? ; Число записаних байтів

.code
main PROC
INVOKE CreateFile,
        ADDR filename, GENERIC_WRITE, DO_NOT_SHARE, NULL,
        CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0
mov fileHandle, eax ; Збережемо дескриптор файлу
.IF eax == INVALID_HANDLE_VALUE
        mov edx, OFFSET errMsg ; Виведемо повідомлення про помилку
call WriteString
jmp QuitNow
.ENDIF
INVOKE WriteFile,
        fileHandle, ; Запишемо текст в файл
        ADDR buffer, ; дескриптор файлу
        bufSize, ; Адреса буфера
        ADDR bytesWritten, ; Число байтів для записування
        0 ; Адреса змінної
        ; Адреса структури OVERLAPPED
        ; не задана
INVOKE CloseHandle, fileHandle ; Закриємо файл

QuitNow:
INVOKE ExitProcess, 0 ; Завершимо програму
main ENDP
END main
```

### Переміщення файлового вказівника

Функція SetFilePointer призначена для переміщення вказівника у відкритому файлі. За допомогою цієї функції можна зробити так, щоб під час записування дані додавалися в кінець файлу, а також організувати доступ до довільних ділянок даних файлу. Прототип функції наведено нижче:

```
SetFilePointer PROTO,
        handle: DWORD, ; дескриптор файлу
        nDistanceLo: SDWORD, ; Число байтів для переміщення
        pDistanceHi: PTR SDWORD, ; Адреса 32-розрядної змінної, яка
        ; містить старше слово 64-розрядного
        ; числа байтів для переміщення
```

moveMethod: DWORD ; Початкова точка для переміщення

Параметр moveMethod визначає відправну точку, відносно якої виконується переміщення вказівника. Він може приймати одне з трьох значень: FILE\_BEGIN, FILE\_CURRENT та FILE\_END. Власне значення, що визначає кількість байтів для переміщення вказівника, є 64-розрядним числом зі знаком, розділеним на 2 частини:

- nDistanceLo- молодші 32-біти;
- pDistanceHi - адреса змінної, яка містить старші 32-біти.

Якщо при виклику функції SetFilePointer параметр pDistanceHi дорівнює нулю, для переміщення файлового вказівника буде використовуватися тільки значення параметра nDistanceLo. Нижче наведено приклад виклику цієї функції для переміщення вказівника в кінець файлу, щоб при подальшій операції записування дані додавалися в його кінець.

```
INVOKE SetFilePointer,  
        fileHandle,          ; дескриптор файлу  
        0,                  ; Молодше значення числа байтів  
        0,                  ; Вказівник на старше значення  
                                ; дорівнює нулю  
FILE_END                      ; Спосіб переміщення - водносно  
                                ; кінця файлу
```

Приклад програми: ReadFile.asm

У програмі ReadFile.asm відкривається текстовий файл, створений при запуску програми WriteFile.asm, потім з нього зчитуються дані, файл закривається і на екрані відображається його вміст:

```
TITLE Використання функції ReadFile (ReadFile.asm)  
INCLUDE Irvine32.inc  
.data  
buffer    BYTE    500    DUP (?)  
bufSize = ($ -buffer)  
errMsg BYTE "Помилка при відкритті файлу", 0dh, 0ah, 0  
filename BYTE "output.txt", 0  
fileHandle DWORD ?      ; дескриптор файлу  
byteCount DWORD ?       ; Число прочитаних байтів  
.code  
main PROC  
INVOKE CreateFile,        ; Відкриємо файл для читання  
        ADDR filename, GENERIC_READ,  
        DO_NOT_SHARE, NULL, OPEN_EXISTING,  
        FILE_ATTRIBUTE_NORMAL, 0  
mov fileHandle, eax       ; Збережемо дескриптор файлу  
.IF eax == INVALID_HANDLE_VALUE  
        mov edx, OFFSET errMsg      ; Виведемо повідомлення про помилку  
        call WriteString  
        jmp QuitNow  
.ENDIF
```

```

    INVOKE ReadFile,      ; Читаємо вміст файлу в буфер
        fileHandle, ADDR buffer, bufSize, ADDR byteCount, 0
    INVOKE CloseHandle,  ; закриємо файл
        fileHandle
    mov esi, byteCount      ; Вставимо нульовий байт в кінець
    mov buffer [esi], 0     ; прочитаного рядка
    mov edx, OFFSET buffer  ; Відобразимо вміст буфера
    call WriteString
QuitNow:
    INVOKE ExitProcess, 0; Завершимо виконання програми
main ENDP
END main

```

## 2. Порядок виконання роботи

1. В сегменті даних опишіть рядок символів, в якому є такі поля: 1) прізвище; 2) ім'я; 3) по батькові; 4) дата народження; 5) місто/село; 6) область; 7) навчальна група (номерів полів не ставити). Перед кожним полем повинна бути довільна (різна) кількість пропусків (пробілів). Після останнього поля теж має бути хоча би один пропуск. В полях пропусків не повинно бути. Всі символи мають бути латинськими. Рядок символів має закінчуватися '00'.

2. В сегменті даних опишіть окремі рядки для кожного поля довжиною, що дорівнює номеру групи, тобто 21/22/23/24/25/26. Опишіть також однобайтові змінні, в яких потрібно буде вказати довжину кожного поля. Опишіть другий рядок символів, довжина якого більша від довжини першого рядка, створеного в пункті 1, на 30 символів. Опишіть третій рядок символів, довжина якого дорівнює довжині першого рядка.

3. **Для роботи з рядками символів використовувати лише ланцюжкові команди!** Складіть підпрограму, яка підраховує і пропускає символи пропуску. Вхідним параметром підпрограми має бути індекс пропуску в рядку, з якого починати пошук. Вихідним параметром має бути індекс першого символа, який не є пропуском.

4. Складіть підпрограму, яка визначає довжину поля. Вхідним параметром підпрограми має бути індекс символа в рядку, з якого починається поле. Вихідним параметром має бути індекс першого пропуску після поля.

5. В головній програмі організуйте пошук і пересилання кожного поля у відповідний рядок, підрахунок кількості символів в кожному полі.

6. Обчисліть загальну кількість пропусків в початковому рядку символів.

7. Перешліть в другий рядок символів кожне поле відповідно до послідовності, вказаної в індивідуальному завданні. Перед кожним полем мають бути пропуски, кількість яких дорівнює номеру поля. Рядок символів має закінчуватися '00'.

8. Для контролю виведіть другий рядок символів на екран.

9. За допомогою функції CreateFile створіть текстовий файл для читання – записування, ім'я файла – ваше прізвище.

10. Запишіть у створений файл спочатку другий, а потім перший рядки символів.

11. Закрийте файл.

12. Відкрийте файл.

13. Організуйте у файлі доступ і прочитайте перший рядок символів (записаний у файлі після другого) в третій рядок.

14. Підрахуйте, скільки разів перший символ (не пропуск) присутній в третьому рядку.

15. Допишіть в кінець файлу назви двох дисциплін (на вибір), які вивчали на першому курсі і оцінки з них.

16. Закрийте файл.

17. Перевірте результат роботи програми.

18. Збережіть програму.

19. У звіті наведіть текст програми, копії вікон зі всіма змінними, а також створений текстовий файл.

### 3. Список літератури

1. Кип Ирвин Язык ассемблера для процессоров Intel 4-е изд 2005.
2. Пирогов В. Ю. Ассемблер для Windows. Изд. 4-е перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 896 с.
3. Юров В.И. Assembler. Учебник для вузов. 2-е изд. — СПб.: Питер, 2010. — 637 с.

### 4. Індивідуальні завдання

ПЗ-21		ПЗ-22		ПЗ-23	
№	Послідовність	№	Послідовність	№	Послідовність
1	П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub>	1	П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>	1	П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub>
2	П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	2	П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	2	П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>
3	П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub>	3	П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub>	3	П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub>
4	П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub>	4	П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub>	4	П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub>
5	П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub>	5	П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub>	5	П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub>
6	П <sub>6</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub>	6	П <sub>6</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub>	6	П <sub>5</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub>
7	П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub>	7	П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>	7	П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub>
8	П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	8	П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	8	П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>
9	П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>1</sub>	9	П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>1</sub>	9	П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub>
10	П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub>	10	П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub>	10	П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub>
11	П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub>	11	П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub>	11	П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub>
12	П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub>	12	П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub>	12	П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub>
13	П <sub>6</sub> , П <sub>3</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>2</sub>	13	П <sub>6</sub> , П <sub>2</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>3</sub>	13	П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>2</sub>
14	П <sub>3</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub>	14	П <sub>2</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>	14	П <sub>7</sub> , П <sub>4</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub>
15	П <sub>4</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	15	П <sub>4</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	15	П <sub>4</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>



16	П <sub>5</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>4</sub>	16	П <sub>7</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>4</sub>	16	П <sub>6</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub>
17	П <sub>1</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>4</sub> , П <sub>5</sub>	17	П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>7</sub>	17	П <sub>1</sub> , П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>6</sub>
18	П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>1</sub>	18	П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>1</sub>	18	П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>6</sub> , П <sub>1</sub>
19	П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>7</sub>	19	П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>5</sub>	19	П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>3</sub>
20	П <sub>6</sub> , П <sub>3</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>7</sub> , П <sub>2</sub>	20	П <sub>6</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub>	20	П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>2</sub>
21	П <sub>3</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>6</sub>	21	П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>6</sub>	21	П <sub>7</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub>
22	П <sub>7</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub>	22	П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub>	22	П <sub>3</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>
23	П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>7</sub>	23	П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>5</sub>	23	П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>3</sub>
24	П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>5</sub>	24	П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub>	24	П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>3</sub> , П <sub>6</sub>
25	П <sub>4</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>1</sub>	25	П <sub>4</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub>	25	П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>1</sub>
26	П <sub>2</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub>	26	П <sub>3</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub>	26	П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub>
27	П <sub>6</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub>	27	П <sub>6</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub>	27	П <sub>5</sub> , П <sub>7</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>2</sub>
28	П <sub>3</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>6</sub>	28	П <sub>2</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>6</sub>	28	П <sub>7</sub> , П <sub>2</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>5</sub>
29	П <sub>2</sub> , П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>3</sub>	29	П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>2</sub>	29	П <sub>2</sub> , П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>7</sub>
30	П <sub>5</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>7</sub> , П <sub>6</sub> , П <sub>3</sub> , П <sub>2</sub>	30	П <sub>7</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>2</sub> , П <sub>3</sub>	30	П <sub>6</sub> , П <sub>1</sub> , П <sub>4</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>2</sub>

ПЗ-24		ПЗ-25		ПЗ-26	
№	Послідовність	№	Послідовність	№	Послідовність
1	П <sub>6</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub>	1	П <sub>4</sub> , П <sub>5</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub>	1	П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub>
2	П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub>	2	П <sub>5</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub>	2	П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub>
3	П <sub>7</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>5</sub>	3	П <sub>2</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>5</sub>	3	П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>5</sub>
4	П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>5</sub> , П <sub>7</sub>	4	П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>2</sub>	4	П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub>
5	П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub>	5	П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>2</sub> , П <sub>3</sub>	5	П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub>
6	П <sub>3</sub> , П <sub>6</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub>	6	П <sub>6</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub>	6	П <sub>4</sub> , П <sub>1</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub>
7	П <sub>6</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub>	7	П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub>	7	П <sub>1</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub>
8	П <sub>7</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub>	8	П <sub>2</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub>	8	П <sub>3</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub>
9	П <sub>5</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub>	9	П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>2</sub>	9	П <sub>5</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>3</sub>
10	П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>5</sub>	10	П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub>	10	П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>5</sub>
11	П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>4</sub>	11	П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>3</sub>	11	П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>6</sub>
12	П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>1</sub>	12	П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub>	12	П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>7</sub>
13	П <sub>3</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>5</sub> , П <sub>4</sub> , П <sub>1</sub> , П <sub>2</sub>	13	П <sub>6</sub> , П <sub>4</sub> , П <sub>2</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>1</sub>	13	П <sub>4</sub> , П <sub>1</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>6</sub> , П <sub>7</sub> , П <sub>2</sub>
14	П <sub>6</sub> , П <sub>4</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub>	14	П <sub>4</sub> , П <sub>3</sub> , П <sub>5</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub>	14	П <sub>1</sub> , П <sub>6</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub>
15	П <sub>4</sub> , П <sub>5</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>2</sub> , П <sub>3</sub> , П <sub>6</sub>	15	П <sub>3</sub> , П <sub>5</sub> , П <sub>2</sub> , П <sub>7</sub> , П <sub>1</sub> , П <sub>6</sub> , П <sub>4</sub>	15	П <sub>6</sub> , П <sub>5</sub> , П <sub>3</sub> , П <sub>7</sub> , П <sub>2</sub> , П <sub>4</sub> , П <sub>1</sub>

16	$\Pi_5, \Pi_7, \Pi_1, \Pi_2, \Pi_3, \Pi_6, \Pi_4$	16	$\Pi_5, \Pi_2, \Pi_7, \Pi_1, \Pi_6, \Pi_4, \Pi_3$	16	$\Pi_5, \Pi_3, \Pi_7, \Pi_2, \Pi_4, \Pi_1, \Pi_6$
17	$\Pi_7, \Pi_1, \Pi_2, \Pi_3, \Pi_6, \Pi_4, \Pi_5$	17	$\Pi_2, \Pi_7, \Pi_1, \Pi_6, \Pi_4, \Pi_3, \Pi_5$	17	$\Pi_3, \Pi_7, \Pi_2, \Pi_4, \Pi_1, \Pi_6, \Pi_5$
18	$\Pi_1, \Pi_2, \Pi_3, \Pi_6, \Pi_4, \Pi_5, \Pi_7$	18	$\Pi_7, \Pi_1, \Pi_6, \Pi_4, \Pi_3, \Pi_5, \Pi_2$	18	$\Pi_7, \Pi_2, \Pi_4, \Pi_1, \Pi_6, \Pi_5, \Pi_3$
19	$\Pi_2, \Pi_3, \Pi_6, \Pi_4, \Pi_5, \Pi_7, \Pi_1$	19	$\Pi_1, \Pi_6, \Pi_4, \Pi_3, \Pi_5, \Pi_2, \Pi_7$	19	$\Pi_2, \Pi_4, \Pi_1, \Pi_6, \Pi_5, \Pi_3, \Pi_7$
20	$\Pi_3, \Pi_6, \Pi_4, \Pi_5, \Pi_7, \Pi_1, \Pi_2$	20	$\Pi_6, \Pi_4, \Pi_3, \Pi_5, \Pi_2, \Pi_7, \Pi_1$	20	$\Pi_4, \Pi_1, \Pi_6, \Pi_5, \Pi_3, \Pi_7, \Pi_2$
21	$\Pi_6, \Pi_1, \Pi_5, \Pi_7, \Pi_4, \Pi_2, \Pi_3$	21	$\Pi_4, \Pi_7, \Pi_5, \Pi_2, \Pi_3, \Pi_1, \Pi_6$	21	$\Pi_1, \Pi_7, \Pi_5, \Pi_3, \Pi_6, \Pi_2, \Pi_4$
22	$\Pi_1, \Pi_5, \Pi_7, \Pi_4, \Pi_2, \Pi_3, \Pi_6$	22	$\Pi_7, \Pi_5, \Pi_2, \Pi_3, \Pi_1, \Pi_6, \Pi_4$	22	$\Pi_7, \Pi_5, \Pi_3, \Pi_6, \Pi_2, \Pi_4, \Pi_1$
23	$\Pi_5, \Pi_7, \Pi_4, \Pi_2, \Pi_3, \Pi_6, \Pi_1$	23	$\Pi_5, \Pi_2, \Pi_3, \Pi_1, \Pi_6, \Pi_4, \Pi_7$	23	$\Pi_5, \Pi_3, \Pi_6, \Pi_2, \Pi_4, \Pi_1, \Pi_7$
24	$\Pi_7, \Pi_4, \Pi_2, \Pi_3, \Pi_6, \Pi_1, \Pi_5$	24	$\Pi_2, \Pi_3, \Pi_1, \Pi_6, \Pi_4, \Pi_7, \Pi_5$	24	$\Pi_3, \Pi_6, \Pi_2, \Pi_4, \Pi_1, \Pi_7, \Pi_5$
25	$\Pi_4, \Pi_2, \Pi_3, \Pi_6, \Pi_1, \Pi_5, \Pi_7$	25	$\Pi_3, \Pi_1, \Pi_6, \Pi_4, \Pi_7, \Pi_5, \Pi_2$	25	$\Pi_6, \Pi_2, \Pi_4, \Pi_1, \Pi_7, \Pi_5, \Pi_3$
26	$\Pi_2, \Pi_3, \Pi_6, \Pi_1, \Pi_5, \Pi_7, \Pi_4$	26	$\Pi_1, \Pi_6, \Pi_4, \Pi_7, \Pi_5, \Pi_2, \Pi_3$	26	$\Pi_2, \Pi_4, \Pi_1, \Pi_7, \Pi_5, \Pi_3, \Pi_6$
27	$\Pi_3, \Pi_6, \Pi_1, \Pi_5, \Pi_7, \Pi_4, \Pi_2$	27	$\Pi_6, \Pi_4, \Pi_7, \Pi_5, \Pi_2, \Pi_3, \Pi_1$	27	$\Pi_4, \Pi_1, \Pi_7, \Pi_5, \Pi_3, \Pi_6, \Pi_2$
28	$\Pi_6, \Pi_2, \Pi_5, \Pi_7, \Pi_4, \Pi_1, \Pi_3$	28	$\Pi_4, \Pi_1, \Pi_5, \Pi_2, \Pi_3, \Pi_7, \Pi_6$	28	$\Pi_1, \Pi_2, \Pi_5, \Pi_3, \Pi_6, \Pi_7, \Pi_4$
29	$\Pi_2, \Pi_5, \Pi_7, \Pi_4, \Pi_1, \Pi_3, \Pi_6$	29	$\Pi_1, \Pi_5, \Pi_2, \Pi_3, \Pi_7, \Pi_6, \Pi_4$	29	$\Pi_2, \Pi_5, \Pi_3, \Pi_6, \Pi_7, \Pi_4, \Pi_1$
30	$\Pi_5, \Pi_7, \Pi_4, \Pi_1, \Pi_3, \Pi_6, \Pi_2$	30	$\Pi_5, \Pi_2, \Pi_3, \Pi_7, \Pi_6, \Pi_4, \Pi_1$	30	$\Pi_5, \Pi_3, \Pi_6, \Pi_7, \Pi_4, \Pi_1, \Pi_2$