|  |
| --- |
| **Assembly Language Team Project Final Result** |

|  |  |  |  |
| --- | --- | --- | --- |
| **제출일** | **2016.6.6** | | |
| **학부** | **컴퓨터공학부** | **컴퓨터공학부** | **수학과** |
| **학번** | **20123782** | **20125753** | **20142052** |
| **이름** | **김남웅** | **박준영** | **정연지** |
| **분반** | **02** | | |
| **담당교수** | **최광남 교수님** | | |



**Practice 11**

Summit the source codes of yours program

|  |  |
| --- | --- |
| TITLE Project  .model small  .stack 100h  .386  loopCount = 20;  .data  buffer BYTE loopCount + 1 DUP(0)  buffer2 BYTE loopCount + 1 DUP(0)  buffer4 BYTE "Error"  Count BYTE 0  .code  main PROC  mov ax, @data  mov ds, ax |  |
| mov ecx, loopCount  call Password  call Crlf | 키를 입력 받는 PasswordPROC 실행 |
| mov ecx, SIZEOF buffer  call Error | 숫자인지 검사하는 ErrorPROC 실행 |
| cmp count,13  jne L2 | ErrorPROC에서 count에 저장한 숫자의수가  주민등록번호의수(13)와 같지않으면 L2로 이동 |
| mov cx,SIZEOF buffer  call Prime |  |
| mov cx, SIZEOF buffer2  mov dx, OFFSET buffer2  jmp L3 | `buffer2에 저장된 소수 값을 출력한다. |
| L2:  mov cx, SIZEOF buffer4  mov dx, OFFSET buffer4  L3:  call WriteString  call Crlf  .exit | Error를 출력한다.  화면에 출력하는 PROC이다. |
| main ENDP  Password PROC  mov esi, 0  L1: |  |
| mov ah, 7  int 21h  cmp al, 0dh  je L4 | input값을 echo없이 받아들인다.  enter키 입력 시 L4로 이동 |
| cmp al, "-"  je L2  mov buffer[esi], al | ‘-’키 입력 시 화면에 출력하기 위해  L2로 이동  ‘-’키가 아니면 al값을  buffer에 저장 |
| mov ah, 2  mov dl, "\*"  int 21h  jmp L3 | practice11 passwordPROC이용  입력 받은 값과 상관없이 ‘\*'값 출력 |
| L2 :  mov ah, 2  mov dl, "-"  int 21h  jmp L1 | ‘-’값 입력 시 화면에 ‘-’출력 |
| L3:  inc esi  loop L1 | enter값이 입력될 때까지 loop문 실행 |
| L4:  ret | enter값 입력 시 passwordPROC 종료 |
| Password ENDP  Prime PROC  mov esi,0  mov ebx,0  L1: |  |
| mov al, buffer[esi]  cmp al, "2"  je L2  cmp al, "3"  je L2  cmp al, "5"  je L2  cmp al, "7"  jne L3 | 소수(2,3,5,7)이 buffer값에 입력 됐을 시  L2로 이동  소수 값이 아니면 L3로 이동 |
| L2 :  mov [buffer2+ebx], al  inc ebx | 소수인 수를 buffer2에 저장한다. |
| L3:  inc esi  loop L1  ret | buffer의 size만큼 loop문 실행 |
| Prime ENDP  Error PROC  mov esi, 0  mov ebx, 0  mov count,0  L1: |  |
| mov al, buffer[esi]  cmp al, 39h  ja L2  cmp al, 30h  jb L2 | 숫자의 아스키 값 범위(30h~39h)를 벗어나면  L2로 이동 |
| inc ebx | ebx에 buffer에 있는 숫자의 수를 저장한다. |
| L2 :  inc esi  loop L1  mov count,bl | 숫자의 수를 count에 저장한다. |
| ret  Error ENDP |  |
| Crlf PROC  mov ah, 2  mov dl, 0ah  int 21h  mov ah, 2  mov dl, 0dh  int 21h  ret  Crlf ENDP | irvine16에 있는 Crlf를 구현 |
| WriteString PROC  mov ah, 40h  mov bx, 1  int 21h  ret  WriteString ENDP | irvine16에 있는 WriteString을 구현 |
| END main |  |

**Final report**

1. **Design of program**

프로시저 3개를 디자인하여 프로그램을 실행한다.

PASSWORD, ERROR, PRIME PROC

1. PASWORD PROC

처음에 입력 받은 값을 buffer에 저장하고 ‘\*’를 출력한다.

‘-‘값은 출력하고 buffer에 넣지 않는다.

Enter키 입력 시 PROC 종료

1. ERROR PROC

숫자의 아스키 값 범위(30h ~ 39h)인지 검사한다.

Buffer에 있는 size만큼 검사를 실행한다.

count변수에 buffer에 들어있는 숫자의 수를 저장한다.

(주민등록번호의 수(13)과 비교하기 위해)

1. PRIME PROC

소수(2,3,5,7)이 buffer에 있을 시 그 값을 buffer2에 저장한다.

Buffer에 있는 size 만큼 검사한다.

main 슈도 코드

충분한 loopCount(최소 14)만큼 password PROC를 통해 입력 받는다.

Error PROC를 통해 입력 받은 값의 숫자의 수를 count에 저장한다.

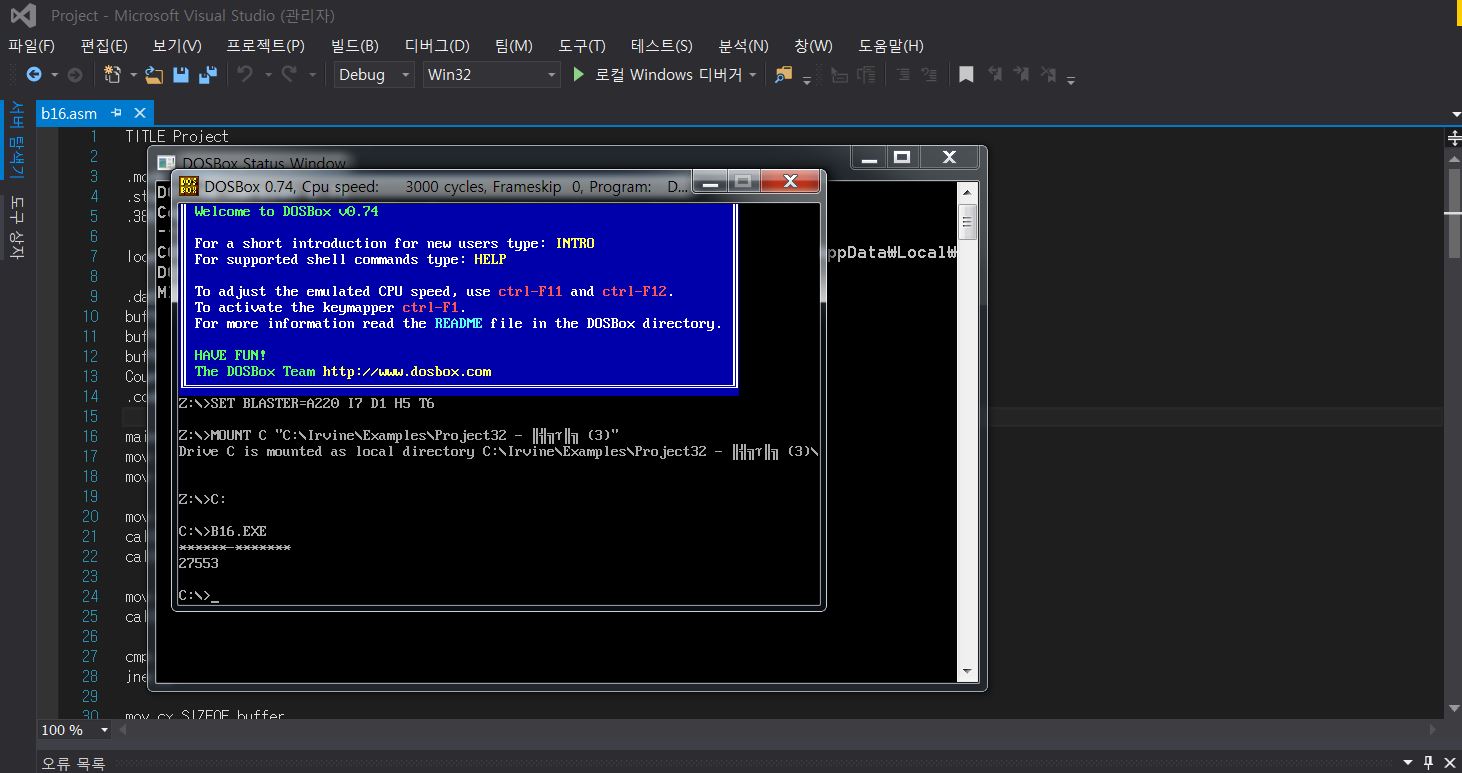
Count의 수가 주민등록번호의 수(13)와 비교한다.

다를 경우 Error를 출력. 프로그램 종료

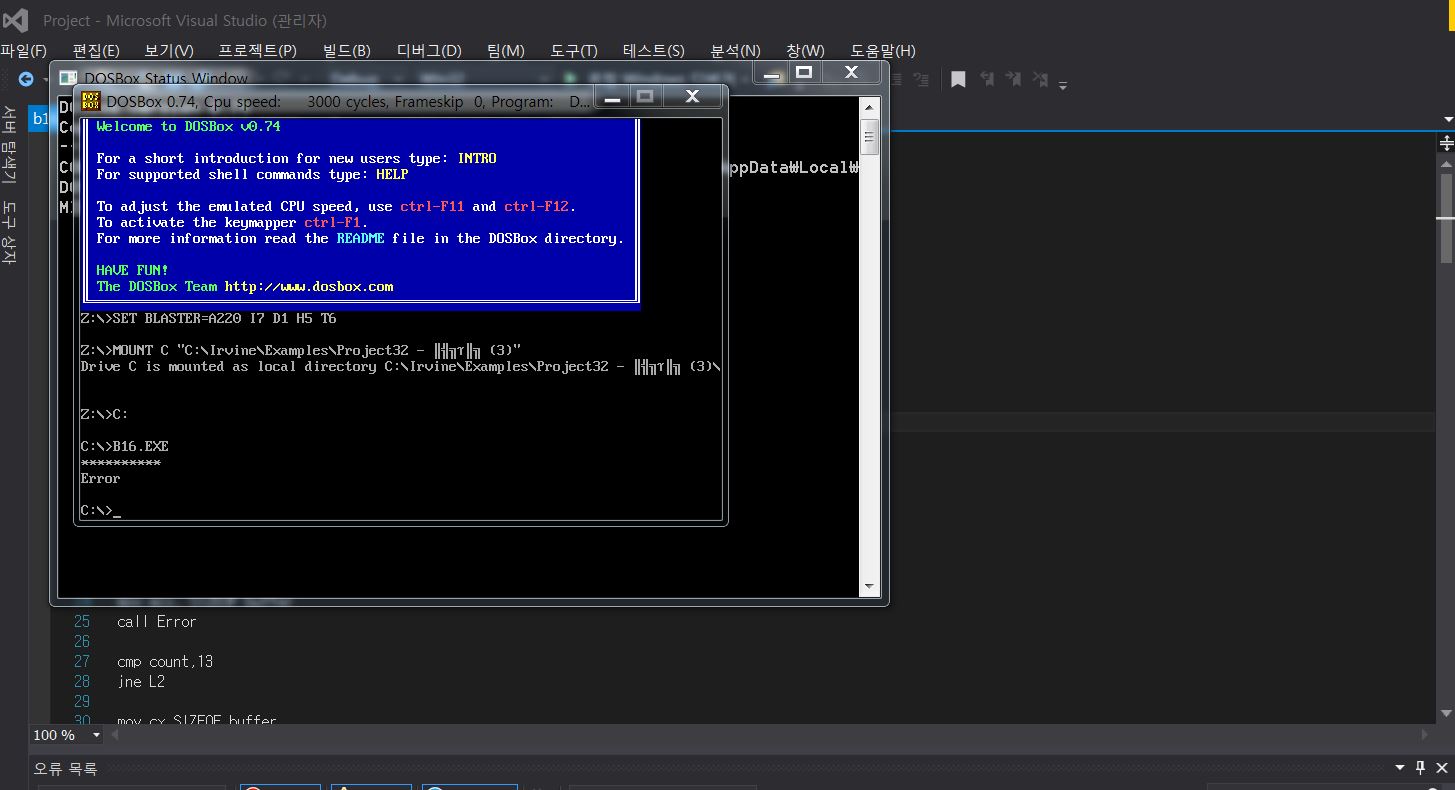
같을 경우 prime PROC를 통해 소수 값을 구별한다.

소수 값을 출력 후 프로그램 종료.

1. **Result of program**

****

정상 결과 출력



오류 결과 출력

1. **The spending time**

**회의 시간 1시간**

**코드 작성 2시간**

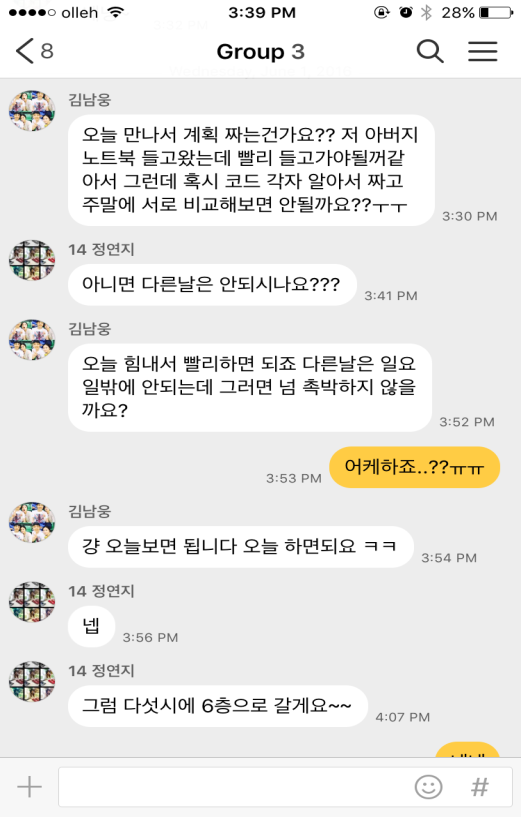
**보고서 작성 1시간**

**총 4시간 소요**

|  |  |
| --- | --- |
| Team member | Role |
| 김남웅 조장(20123782) | 보고서 작성, 코드 작성 |
| 박준영(20125753) | 코드 작성, FSM 작성 |
| 정연지(20142052) | 코드 작성, FSM 작성 |

1. **Role and contribution of each member**

1. **Team Meeting Pictures or Capture image of Kakao talk, etc**

****

1. **Team’s thoughts on this project**

최초의 계획은 FSM대로 코드를 작성하려 하였으나 그렇게 하면 효율적이지 못하다고 생각했습니다. 그래서 loop를 돌려서 각각의 상황에서 예외를 만들어서 error를 출력하게 하는 방법으로 프로그램을 작성하였습니다. Proc를 3개 만들어 각각의 상황에 맞게 사용하였고 원하는 요구사항을 충족할 수 있었습니다. 팀 프로젝트를 하면서 만나는 과정이나 역할분담에서 많은 고생이 있었는데 이런 과정을 겪으면서 실제 사회에서 프로젝트를 간접적으로 체험할 수 있는 계기가 되어서 좋은 경험이 됐습니다.