Задание 7

7.1.

Пусть есть описания const n = 400; type mat = array[1..n, 1..n] of word;

Реализовать на ассемблере функцию со стандартным соглашением о связях

```
Function prl (var A: mat; n, m: dword): word;
begin prl := A[n, m-1] end;
```

Считать, что m будет передано корректно: $2 \leq m \leq n$.

```
.data
  N equ 4
  x dw N*N dup(?)
.code
func proc
  push ebp
  mov ebp, esp
  push edx
  push ecx
  push ebx
  mov ebx, [ebp+8]; offset x
  mov ecx, [ebp+12]; N
  mov eax, ecx
  dec ecx
  mul ecx
  add eax, [ebp+16]
  dec eax
  mov ax, [ebx+eax*2-2]
```

```
  pop ebx
  pop ecx
  pop edx
  pop ebp
  ret 4*3
func endp
Start:
    push M
    push N
    push offset x
    call func
    outwordln ax
```

7.2.

Описать на Ассемблере процедуру D5 (n,d), которая записывает в d значение старшей цифры в записи числа n в 5-чной системе счисления. Параметры: n - двойное слово (dd), число без знака, передаётся по ссылке. Процедура должна удовлетворять стандартным соглашениям о связях. Выписать пример вызова процедуры, описав необходимые переменные.

```
.data
   n dd ?
   d db ?
.code
D5 proc
  push ebp
  mov ebp, esp
  push eax
```

```
  push edx
  push ecx
  push ebx
  mov eax, [ebp+8] ;N
 mov eax, [eax]
  mov ebx, 5
L:cmp eax, 5
  jb M
  cdq
  div ebx
  jmp L
M:mov edx, [ebp+12]
  mov [edx], al
  pop ebx
  pop ecx
  pop edx
  pop eax
  pop ebp
  ret 4*2
D5 endp
Start:
    push offset d
    push offset n
    call D5
```

7.3.

Описать на Ассемблере процедуру D7 (n,d), которая записывает в d значение старшей цифры в записи числа n в 7-чной системе счисления. Параметры: n - двойное слово (dd), число без знака, передаётся по ссылке. Процедура должна удовлетворять стандартным соглашениям о связях. Выписать пример вызова процедуры, описав необходимые переменные.

```
.data
  n dd ?
  d db ?
.code
D7 proc
  push ebp
  mov ebp, esp
  push eax
  push edx
  push ecx
  push ebx
  mov eax, [ebp+8] ;N
  mov eax, [eax]
  mov ebx, 7
L:cmp eax, 7
  jb M
  cdq
  div ebx
  jmp L
M:mov edx, [ebp+12]
  mov [edx], al
```

```
  pop ebx
  pop ecx
  pop edx
  pop eax
  pop ebp
  ret 4*2
D7 endp
Start:
    push offset d
    push offset n
    call D7
```

7.4.

Пусть есть описания на языке Free Pascal

```
Type T = record P: longint; Q : char end ;
Procedure inint (var X :T) ; begin X.P:=2;  X.Q:= ‘(‘ end;
```

Пусть тип T описан на Ассемблере в виде структуры

```
T struc
    P dd ?
    Q db ?
T ends
```

Реализовать процедуру init на Ассемблере. Процедура должна удовлетворять стандартным соглашениям о связях.

```
T struc
  P dd ?
  Q db ?
T ends
.data
 x T <>
.code
D proc
 push ebp
 mov ebp, esp
 push eax
 mov eax, [ebp+8]
 mov dword ptr [eax], 2
 mov byte ptr [eax+4], '('
 pop eax
```

```
  pop ebp
  ret 4
D endp
Start:
    push offset x
    call D
```

7.5.

Пусть есть описания на Паскале:

Type sbyte = -128..127; nabs = array[0..36] of sbyte;

Написать на Ассемблере процедуру с заголовком

Procedure Subtr (var a:longword; var x:nubs; N:longword (длина массива));

Реализующую действие $a := a - \Sigma x_i$. Процедура должна удовлетворять стандартным соглашениям о связях.

```
.data
   N equ 36
   x dw N dup(?)
   a dd ?
.code
D proc
  push ebp
  mov ebp, esp
  push eax
  push ebx
  push ecx
  push edx
  mov ebx, [ebp+12]; offset x
```

```
  mov ecx, [ebp+16]; N
  xor eax, eax
L:movsx edx, word ptr[ebx+ecx*2-2]
  add eax, edx
  Loop L
  mov ebx, [ebp+8]
  sub [ebx], eax
  pop edx
  pop ecx
  pop ebx
  pop eax
  pop ebp
  ret 3*4
D endp
Start:
    mov ecx, N
M:   inint x[ecx*2-2]
    Loop M
    push N
    push offset x
    push offset a
    call D
```

7.6.

Описать на Ассемблере процедуру MaxNom (x,n), здесь x – массив из n чисел со знаком формата слова (dw). Процедура печатает наибольший элемент массива и его номер. Элементы массива нумеруются, начиная с 1 (x_1, x_2 ...). Процедура должна удовлетворять стандартным соглашениям о связях. Когда наибольших элементов несколько, печатать номер первого из них.

```
.data
  N equ 4
  x dw N dup(?)
.code
MaxNom proc
  push ebp
  mov ebp, esp
  push eax
  push ebx
  push ecx
  push edx
  push esi
  mov ebx, [ebp+8]
  mov ecx, [ebp+12]
  mov ax, -128
L:mov dx, [ebx+ecx*2-2]
  cmp ax, dx
  jge CNT
  mov ax, dx
  mov esi, ecx
CNT:Loop L
```

```
  mov ecx, [ebp+12]
  inc ecx
  sub ecx, esi
  outintln ax
  outwordln ecx
  pop esi
  pop edx
  pop ecx
  pop ebx
  pop eax
  pop ebp
  ret 3*4
MaxNom endp
Start:
    mov ecx, N
M:  inint x[ecx*2-2]
    Loop M
    push N
    push offset x
    call MaxNom
```

7.7.

Реализовать на языке Ассемблера процедуру, которая на языке Free Pascal имеет вид:

```
Procedure Change (var n:integer);
begin if n < 10 then n := n+1 else n:= n-1 end;
```

Здесь n – число со знаком. Процедура должна удовлетворять стандартным соглашениям о связях. Привести пример вызова этой процедуры.

```
  n dw ?
.code
D proc
  push ebp
  mov ebp, esp
  push eax
  push ebx
  mov ebx, [ebp+8]
  mov ax, [ebx]
  cmp ax, 10
  jl L
  dec ax
  jmp CN
L: inc ax
CN:mov [ebx], ax
  pop ebx
  pop eax
  pop ebp
  ret 4
```

```
D endp
Start:
    inint n
    push offset n
    call D
```

7.8. Описать на Ассемблере процедуру MaxNom(x,n), здесь x – массив из n чисел со знаком формата T, где T **equ db**, при № **mod** 3=0, T **equ dw**, при № **mod** 3=1 и T **equ dd**, при № **mod** 3=2, где № – Ваш номер в ведомости. Процедура печатает наибольший элемент массива и его номер. Элементы массива нумеруются, начиная с 1 (x_1, x_2, …). Процедура должна удовлетворять стандартным соглашениям о связях. Когда наибольших элементов несколько, печатать номер первого из них.

```
.data
  N equ 4
  x dw N dup(?)
.code
MaxNom proc
 push ebp
 mov ebp, esp
 push eax
 push ebx
 push ecx
 push edx
 push esi
 mov ebx, [ebp+8]
 mov ecx, [ebp+12]
 mov ax, -128
```

```
L:mov dx, [ebx+ecx*2-2]
  cmp ax, dx
  jge CNT
  mov ax, dx
  mov esi, ecx
CNT:Loop L
  mov ecx, [ebp+12]
  inc ecx
  sub ecx, esi
  outintln ax
  outwordln ecx
  pop esi
  pop edx
  pop ecx
  pop ebx
  pop eax
  pop ebp
  ret 3*4
MaxNom endp
Start:
    mov ecx, N
M:   inint x[ecx*2-2]
    Loop M
    push N
    push offset x  call MaxNom
```

7.9. Пусть есть описание const n = 4320; type vec=array[1..n] of word;

Реализовать на Ассемблере функцию со стандартными соглашениями о связях

```
Function L2(var A:vec; n:dword): word;
Begin L2:=A[n] end;
```

Смысл параметров n – длина массива A

```
L2 proc
    Push ebp
    Mov ebp, esp
    Push ebx
    Push ecx
    Mov ebx, [ebp+12]; n
    Mov ecx, [ebp+8]; offset A
    Mov eax, [ecx+2*ebx-2]
    Pop ecx
    Pop ebx
    Pop ebp
    Ret 8
L2 endp
```

7.10. Пусть есть описание const n = 1230; type vec=array[1..n] of word;

Реализовать на Ассемблере функцию со стандартными соглашениями о связях

```
Function L2(var A:vec; n:dword): word;
Begin L2:=A[n] end;
```

Смысл параметров n – длина массива A

```
L2 proc
    Push ebp
    Mov ebp, esp
    Push ebx
    Push ecx
    Mov ebx, [ebp+12]; n
    Mov ecx, [ebp+8]; offset A
    Mov eax, [ecx+4*ebx-4]
    Pop ecx
    Pop ebx
    Pop ebp
    Ret 8
L2 endp
```