```
int get_digit (zip_dig z, int dig) {
   return z[dig];
}
```

```
; edx = z
; eax = dig
mov eax, dword [edx+4*eax] # z[dig]
```

```
void zincr(zip_dig z) {
 int i;
 for (i = 0; i < ZLEN; i++)
  z[i]++;
}
```

```
                                ;  edx = z
 mov eax, 0                     ;  eax = i
.L4:                            ; loop:
 add dword [edx + 4 * eax], 1   ;  z[i]++
 add eax, 1                     ;  i++
 cmp eax, 5                     ;  i vs. 5
 jne .L4                        ;  if (!=) goto loop
```

```
int *get_pgh_zip(int index){
  return pgh[index];
}
```

```
      ; eax = index
     lea eax, [eax + 4 * eax] ; 5 * index
     lea eax, [pgh + 4 * eax] ; pgh + (20 * index)
```

```
int get_pgh_digit (int index, int dig)
{
  return pgh[index][dig];
}
```

```
      mov   eax, dword [ebp + 8]        ; index
      lea   eax, [eax + 4 * eax]        ; 5*index
      add   eax, dword [ebp + 12]       ; 5*index+dig
      mov   eax, dword [pgh + 4 * eax]  ; смещение
                                        ;4*(5*index+dig)
```

Доступ к элементу Mem[Mem[univ+4*index]+4*dig]

```
int get_univ_digit (int index,int dig)
{
  return univ[index][dig];
}
```

```
      mov   eax, dword [ebp + 8]        ; index
      mov   edx, dword [univ + 4 * eax] ; p = univ[index]
      mov   eax, dword [ebp + 12]       ; dig
      mov   eax, dword [edx + 4 * eax]  ; p[dig]
```

```
/* Получение элемента  a[i][j] */
int var_ele(int n,int a[n][n],int i,int j)
{
  return a[i][j];
}
```

```
      mov   eax, dword [ebp + 8]  ; n
      sal   eax, 2                ; n*4
      mov   edx, eax              ; n*4
      imul  edx, dword [ebp + 16] ; i*n*4
      mov   eax, dword [ebp + 20] ; j
      sal   eax, 2                ; j*4
      add   eax, dword [ebp + 12] ; a + j*4
      mov   eax, dword [eax + edx]; *(a + j*4 + i*n*4)
```

```
/* Выборка столбца j из массива */
void fix_column
  (fix_matrix a, int j, int *dest)
{
  int i;
  for (i = 0; i < N; i++)
    dest[i] = a[i][j];
}
```

```
;ecx = ajp
;ebx = dest
;edx = i
.L8:                                        ; loop:
          mov  eax, dword [ecx]             ; считываем *ajp
          mov  dword [ebx + 4 * edx], eax   ; сохраняем в dest[i]
          add  edx, 1                       ; i++
          add  ecx, 64                      ; ajp += 4*N
          cmp  edx, 16                      ; i vs. N
          jne  .L8                          ; if !=, goto loop
```

```
/* Выборка столбца j из массива */
void var_column
  (int n, int a[n][n],
   int j, int *dest)
{
  int i;
  for (i = 0; i < n; i++)
    dest[i] = a[i][j];
}
```

```
;ecx = ajp
;edi = dest
;edx = i
;ebx = 4*n
;esi = n

.L18:                                       ; loop:
   mov eax, dword [ecx]                     ; считываем *ajp
          mov  dword [edi + 4 * edx], eax   ; сохраняем в dest[i]
          add  edx, 1                       ; i++
          add  ecx, ebx                     ; ajp += 4*n
          cmp  esi, edx                     ; n vs. i
          jg   .L18                         ; if (>) goto loop
```

```
/* Выборка столбца j из массива */
void var_column
  (int n, int a[n][n],
   int j, int *dest)
{
  int i;
  dest--;
  for (i = n; i != 0; i--)
    dest[i] = a[i-1][j];
}
```

```
;ecx = a+4*n*(n-1)+4*j
;edi = dest – 4
;edx = n
;ebx = 4*n
;esi = освободился

.L18:                                   ; loop:
  mov  eax, dword [ecx]                 ; считываем *(ajp+…)
          mov  dword [edi + 4 * edx], eax   ; сохраняем в dest[i]
          sub   ecx, ebx                ; ajp -= 4*n
          sub   edx, 1                  ; i--
          jnz    .L18                   ; if (!=) goto loop
```