

a) dynamické 2D pole ve vektoru

```
int n;  
int i,j;  
int *A;  
  
n=5;  
A=(int *)malloc(n*n*sizeof(int));  
*(A+n*i+j)=....
```

b) dynamické 2D pole ve vektoru s pomocným vektorem ukazatelů

```
typedef int *vec;  
int n;  
int i,j;  
int *A;  
vec *M;  
  
n=5;  
M=(vec *)malloc(n*sizeof(vec));  
A=(int *)malloc(n*n*sizeof(int));  
for (i=0;i<n;i++) M[i]=A+n*i;  
M[i][j]=....
```

c) dynamické 2D pole s pevným počtem sloupců

```
typedef int V[6];  
  
V *mat;  
int i,j;  
int n;  
  
n=5;  
mat=(V *)malloc(n*sizeof(V));  
mat[i][j]=....
```

d) dynamické 2D pole s vektorem ukazatelů na řádky a alokací paměti pro každý řádek zvlášť

```
typedef int *vec;  
int n;  
int i,j;  
vec *M;  
  
n=5;  
M=(vec *)malloc(n*sizeof(vec));  
for (i=0;i<n;i++) M[i]=(int *)malloc(n*sizeof(int));  
M[i][j]=....
```