2) Rewrite the matrix multiplication function using dynamically allocated arrays. The header for your function should be
void mult(int **a, int **b, int **c, int rows) where each matrix is a rows x rows matrix.

```
void mult(int **a, int **b, int **c, int rows)
{
  MALLOC(c, rows*sizeof(*c));
  int i,j,K;
  for(i=0; i<rows; i++)
    CALLOC(c[i], rows, sizeof(**c));
  for(i=0; i<rows; i++)
    for(j=0; j<rows; j++)
      for(K=0; K<rows; K++)
        c[i][j] += a[i][K] * b[K][j];
}
```

5)
```
void transpose(int a[][MAX_SIZE])
{
  int i,j,temp;
  for(i=0; i<MAX_SIZE-1; i++)
    for(j=i+1; j<MAX_SIZE; j++)
      SWAP(a[i][j], a[j][i], temp);
}
```

Inner Loop Invariant

- Just before the iteration when $j=K$, the matrix element at row i, column x has been interchanged with the element at row x, column i $\forall x \in N$ s.t. $i+1 \leq x < K$.

Initialization: $j=i+1$. $\because K=i+1$, the range $i+1 \leq x < K$ doesn't make sense.
$\therefore$ The loop invariant trivially holds

Maintenance: $j=K$. Just before this iteration, let the loop invariant hold.
$\therefore \forall x \in N$, $i+1 \leq x < K$, the element at row x, column i has been interchanged with the element at row i, column x.

- Now, $j=K$. By correctness of SWAP macro, the element at row i, column K has been interchanged with the element at row K, column i.