***Given the base address of an array, find the address of a matrix using both row major and column major***

#include <stdio.h>

int main() {

int a[10][12], r, c;

printf("Assuming base address as 2400 \n");

printf("Enter a[r][c] \n");

printf("a[r]:");

scanf("%d",&r);

printf("\na[c]:");

scanf("%d",&c);

int bp;

//bp = &a[0][0]; //Change to \*bp at line 14 & 11 if bp is defined here

bp = 2400; //Assuming base address as 2400, Change to bp at line 14 & 11

int ba = bp;

printf("\nBase address: %d\n",ba);

int offsetr = r\*12+c;

int offsetc = c\*10+r;

int addressr = offsetr\*4+ba;

int addressc = offsetc\*4+ba;

printf("Offset in row major: %d\n",offsetr);

printf("Offset in column major: %d\n",offsetc);

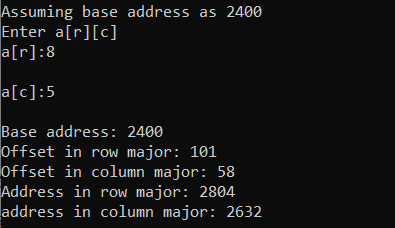
printf("Address in row major: %d\n",addressr);

printf("address in column major: %d\n",addressc);

return 0;

}

Output:



***Given the base address and an actual address of a particular element, find whether the array is stored in row major or column major***

#include <stdio.h>

int main() {

int a[10][12], r, c, ua;

printf("Assuming base address as 2400 \n");

printf("Enter a[r][c] \n");

printf("a[r]:");

scanf("%d",&r);

printf("\na[c]:");

scanf("%d",&c);

int bp;

//bp = &a[0][0]; //Change to \*bp at line 14 & 11 if bp is defined here

bp = 2400; //Assuming base address as 2400, Change to bp at line 14 & 11

int ba = bp;

//printf("\nBase address: %d\n",ba);

int offsetr = r\*12+c;

int offsetc = c\*10+r;

int addressr = offsetr\*4+ba;

int addressc = offsetc\*4+ba;

printf("Enter predicted address of element: ");

scanf("%d",&ua);

if(addressr == ua){

printf("\nMatrix is row-major");

}

else if(addressc == ua){

printf("\nMatrix is column-major");

}

else{

printf("\npredicted address is invalid");

}

return 0;

}

Output:

