

## Dwuwymiarowe tablice dynamiczne w C++ i ANSI-C

## 1. JEDNO- I DWUWYMIAROWE TABLICE DYNAMICZNE – C++

za pomocą **new** i **delete**:

```

#include <iostream>
using namespace std;

int main (void)
{
    //wczytaj wymiary
    int w,k;
    cout<<"Liczba wierszy: ";
    cin>>w;
    cout<<"Liczba kolumn: ";
    cin>>k;

    //////////// 1D ////////////
    cout<<"\n\nTABLICA 1D - C++\n";
    double* tab = new double [w]; //alokacja pamieci
    for ( int i = 0; i < w; ++i) //wpisanie wartosci do tablicy
        tab[i]=i/10.0;

    //wypisz tab[w]
    for ( int i = 0; i < w; ++i)
        cout<<tab[i]<<'\t';
    cout<<endl;

    //zniszcz tab
    delete [] tab; //uwolnienie pamieci
    tab = NULL;

    //////////// 2D ////////////
    cout<<"\n\nTABLICA 2D - C++\n";
    double **tab2 = new double *[w]; //alokacja pamieci
    for ( int i = 0; i < w; ++i )
    {
        tab2[i] = new double [k]; //alokacja pamieci
        for ( int j = 0; j < k; ++j) //wpisanie wartosci do tablicy
            tab2[i][j]=(i+j)/10.0;
    }

    //wypisz tab2[w][k]
    for ( int i = 0; i < w; ++i, cout<<endl )
        for ( int j = 0; j < k; ++j)
            cout<<tab2[i][j]<<'\t';

    //zniszcz tab2
    for ( int i(0); i < w; ++i )
        delete [] tab2[i]; //uwolnienie pamieci
    delete [] tab2; //uwolnienie pamieci
    tab2 = NULL;

    system("PAUSE");
    return 0;
}

```

## 2. DWUWYMIAROWE TABLICE DYNAMICZNE – ANSI-C

za pomocą **malloc** i **free**:

```

#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    //wczytaj wymiary
    int w,k;
    printf("Liczba wierszy: ");
    scanf("%d",&w);
    printf("Liczba kolumn: ");
    scanf("%d",&k);

    //////////// 2D ////////////
    printf("\n\nTABLICA 2D - ANSI C\n");
    double **tab2;
    tab2=(double**)malloc(w*sizeof(double *)); //alokacja pamieci
    for(int i=0; i<w; i++)
    {
        tab2[i]=(double*)malloc(k*sizeof(double));
        for ( int j = 0; j < k; ++j) //wpisanie wartosci do tablicy
            tab2[i][j]=(i+j)/10.0;
    }

    //wypisz tab2[w][k]
    for ( int i = 0; i < w; ++i, printf("\n") )
        for ( int j = 0; j < k; ++j)
            printf("%.1f\t",tab2[i][j]);

    //zniszcz tab2
    for(int i=0; i<w; i++)
        free(tab2[i]); //uwolnienie pamieci
    free(tab2); //uwolnienie pamieci
    tab2 = NULL;

    system("PAUSE");
    return 0;
}

```