

1- Write a program to dynamically allocate memory for an integer and initialize it with the value 5. And using free

اكتب برنامجًا لتخصيص الذاكرة ديناميكيًا لعدد صحيح وقم بتهيئته بالقيمة 5.
And using free

Output

5

Solution

```
// www.gammal.tech
#include <stdio.h>
#include <stdlib.h>

int main() {
    int *x = (int *)malloc(sizeof(int));
    *x = 5;
    printf("%d\n", *x);
    free(x);
    return 0;
}
```

2- Write a program to dynamically allocate memory for two integers and initialize them with values 10 and 20. And using free

اكتب برنامجًا لتخصيص الذاكرة ديناميكيًا لعددتين صحيحتين وتهيئتهما بالقيمتين 10 و20.
And using free.

Output

10 20

Solution

```
// www.gammal.tech

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

int main() {
    int *x = (int *)malloc(2 * sizeof(int));
    x[0] = 10;
    x[1] = 20;
    for (int i = 0; i < 2; i++)
        printf("%d ", x[i]);
    free(x);
    return 0;
}
```

3- Write a program to dynamically allocate memory for three floats and initialize them with values 3.14, 2.71, and 1.618. And using free

اكتب برنامجًا لتخصيص الذاكرة ديناميكيًا 3 floats وقم بتهيئتها بالقيم 3.14،
و2.71، و1.618. And using free

Output

```
3.14 2.71 1.62
```

Solution

```
// www.gammal.tech

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

int main() {
    float *x = (float *)malloc(3 * sizeof(float));
    x[0] = 3.14;
    x[1] = 2.71;
    x[2] = 1.618;
    for (int i = 0; i < 3; i++)
        printf("%.2f ", x[i]);
    free(x);
    return 0;
}
```

4- Write a program to dynamically allocate memory for an array of 4 characters and initialize them with characters 'A', 'B', 'C', and 'D'. And using free

اكتب برنامجًا لتخصيص الذاكرة ديناميكيًا لـ array مكونة من 4 أحرف وتهيئتها بالأحرف "A" و "B" و "C" و "D".

And using free

Output

A B C D

Solution

```
// www.gammal.tech
#include <stdio.h>
#include <stdlib.h>

int main() {
    char *x = (char *)malloc(4 * sizeof(char));
    for (int i = 0; i < 4; i++)
        x[i] = 'A' + i;

    for (int i = 0; i < 4; i++)
        printf("%c ", x[i]);

    free(x);
    return 0;
}
```

5- Write a program to dynamically allocate memory for an array of 4 integers and initialize them with values 1, 3, 5, and 7. And using free

اكتب برنامجًا لتخصيص الذاكرة ديناميكيًا لـ array مكونة من 4 أعداد صحيحة وقيمها 1 و 3 و 5 و 7. And using free

Output

1 3 5 7

Solution

```
// www.gammal.tech

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

int main() {
    int *x = (int *)malloc(4 * sizeof(int));
    for (int i = 0; i < 4; i++)
        x[i] = (i * 2) + 1;

    for (int i = 0; i < 4; i++)
        printf("%d ", x[i]);

    free(x);
    return 0;
}
```

6- Write a program that dynamically allocates memory for an integer array of size 6 using malloc. Initialize the array with values corresponding to twice the index plus one. After printing the initial memory address of the allocated array, free the memory. Then, allocate a new block of memory for the same-sized array, print the new memory address.

اكتب برنامج الذي يخصص الذاكرة ديناميكياً لـ array أعداد صحيحة بحجم 6 باستخدام malloc. قم بتهيئة array بقيم تقابل ضعف الفهرس زائد واحد. بعد طباعة عنوان الذاكرة الأولي لـ array المخصصة، قم مسح الذاكرة. ثم قم بتخصيص كتلة جديدة من الذاكرة لـ array ذات الحجم نفسه، واطبع عنوان الذاكرة الجديد.

Output

Initial Memory Address: 0x9862a0
New Memory Address: 0x9862a0

Solution

```
// www.gammal.tech

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

int main() {
    // Allocate memory for an integer array of size 6
    int *x = (int *)malloc(6 * sizeof(int));

    // Print the initial memory address of the allocated array
    printf("Initial Memory Address: %p\n", (void *)x);

    // Initialize the array with values twice the index plus one
    for (int i = 0; i < 6; i++)
        x[i] = (i + 1) * 2;

    // Free the allocated memory
    free(x);

    // Allocate a new block of memory for the same-sized array
    x = (int *)malloc(6 * sizeof(int));

    // Print the new memory address
    printf("New Memory Address: %p\n", (void *)x);

    return 0;
}
```

7- Write a program that dynamically allocates memory for a double array of size 4 using malloc. Initialize the array with values corresponding to 1.5 times the index plus one. After printing the initial memory address of the allocated array, print the elements of the array separated by spaces. Free the memory, then allocate a new block of memory for the same-sized array and print the new memory address.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا double array الحجم 4 باستخدام malloc. قم بتهيئة array بقيم تقابل 1.5 مرة من الفهرس زائد واحد. بعد طباعة عنوان الذاكرة الأولي array المخصصة، قم بطباعة عناصر array مفصولة بمسافات. قم مسح الذاكرة، ثم قم بتخصيص كتلة جديدة من الذاكرة array ذات الحجم نفسه وطباعة عنوان الذاكرة الجديد.

Output

```
Initial Memory Address: 0xa622a0
Initial Array Elements: 1.500000 3.000000 4.500000 6.000000
New Memory Address: 0xa622a0
```

Solution

```

// www.gammal.tech

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

int main() {
    // Allocate memory for a double array of size 4
    double *x = (double *)malloc(4 * sizeof(double));

    // Print the initial memory address of the allocated array
    printf("Initial Memory Address: %p\n", (void *)x);

    // Initialize the array with values 1.5 times the index plus one
    for (int i = 0; i < 4; i++)
        x[i] = (i + 1) * 1.5;

    // Print the elements of the array separated by spaces
    printf("Initial Array Elements: ");
    for (int i = 0; i < 4; i++)
        printf("%lf ", x[i]);
    printf("\n");

    // Free the allocated memory
    free(x);

    // Allocate a new block of memory for the same-sized array
    x = (double *)malloc(4 * sizeof(double));

    // Print the new memory address
    printf("New Memory Address: %p\n", (void *)x);

    return 0;
}
```

8- Write a program that dynamically allocates memory for an integer array of size 3 using malloc. Initialize the array with values corresponding to three times the index plus one. After printing the initial memory address of the allocated array, free the memory. Then, allocate a new block of memory for the same-sized array and print the new memory address.

اكتب برنامجًا يخصص الذاكرة ديناميكيًا array أعداد صحيحة بالحجم 3 باستخدام malloc. قم بتهيئة array بقيم تقابل ثلاثة أضعاف الفهرس زائد واحد. بعد طباعة عنوان الذاكرة الأولي array المخصصة، قم مسح الذاكرة. ثم قم بتخصيص كتلة جديدة من الذاكرة array ذات الحجم نفسه وطباعة عنوان الذاكرة الجديد.

Output

```
Initial Memory Address: 0x17672a0
New Memory Address: 0x17672a0
```

Solution

```
// www.gammal.tech

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

int main() {
    // Allocate memory for an integer array of size 3
    int *x = (int *)malloc(3 * sizeof(int));

    // Print the initial memory address of the allocated array
    printf("Initial Memory Address: %p\n", (void *)x);

    // Initialize the array with values three times the index plus one
    for (int i = 0; i < 3; i++)
        x[i] = (i + 1) * 3;

    // Free the allocated memory
    free(x);

    // Allocate a new block of memory for the same-sized array
    x = (int *)malloc(3 * sizeof(int));

    // Print the new memory address
    printf("New Memory Address: %p\n", (void *)x);

    return 0;
}
```

9- Write a program that uses calloc to allocate memory for an integer array of size 5. Initialize the array with values 1 to 5, and print the elements. And using free

اكتب برنامجًا يستخدم calloc لتخصيص الذاكرة array أعداد صحيحة بحجم 5. قم بتهيئة array بالقيم من 1 إلى 5، ثم اطبع العناصر. And using free

Output

```
1 2 3 4 5
```

Solution

```
// www.gammal.tech

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

int main() {
    // Allocate memory for an integer array of size 5 using calloc
    int *arr = (int *)calloc(5, sizeof(int));

    // Initialize the array with values 1 to 5
    for (int i = 0; i < 5; i++)
        arr[i] = i + 1;

    // Print the elements of the array
    for (int i = 0; i < 5; i++)
        printf("%d ", arr[i]);

    // Free the allocated memory
    free(arr);

    return 0;
}
```

10- Create a program that dynamically allocates memory for a character array of size 8 using calloc. Initialize the array with the letters 'A' to 'H', and print the characters.

And using free


قم بإنشاء برنامج يقوم بتخصيص الذاكرة ديناميكيًا لمجموعة أحرف بحجم 8 باستخدام calloc. قم بتهيئة array بالأحرف من "A" إلى "H"، ثم قم بطباعة الأحرف. And using free.

Output

```
A B C D E F G H
```


Solution

```


// www.gammal.tech

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

int main() {
    // Allocate memory for a character array of size 8 using calloc
    char *arr = (char *)calloc(8, sizeof(char));

    // Initialize the array with the letters 'A' to 'H'
    for (int i = 0; i < 8; i++)
        arr[i] = 'A' + i;

    // Print the characters of the array
    for (int i = 0; i < 8; i++)
        printf("%c ", arr[i]);

    // Free the allocated memory
    free(arr);

    return 0;
}
```
