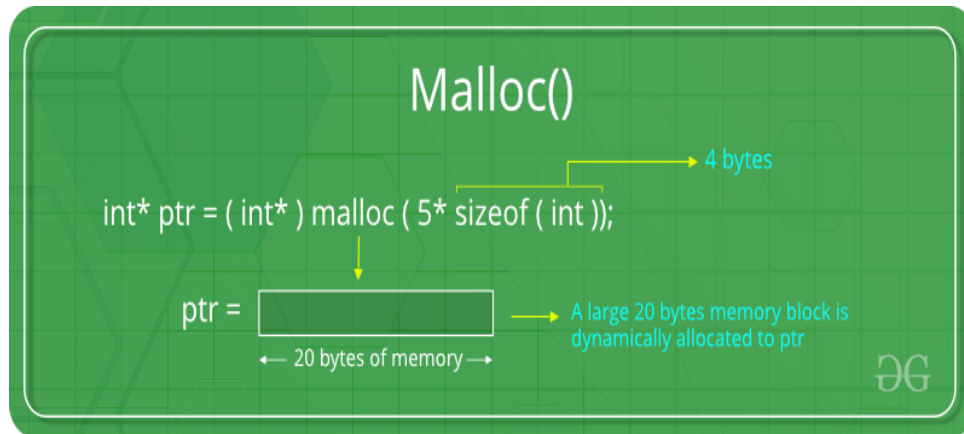


Dynamic Memory Allocation Programs

Example Program for malloc():



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

int main()
{
    int n, i, *ptr, sum = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);
    ptr = (int*) malloc(n * sizeof(int));
    if(ptr == NULL)
    {
        printf("Error! memory not allocated.");
        exit(0);
    }

    printf("Enter elements: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum=sum+ *(ptr+i);
    }

    printf("Sum is: %d\n", sum);
    free(ptr);
    return 0;
}
```

Output:

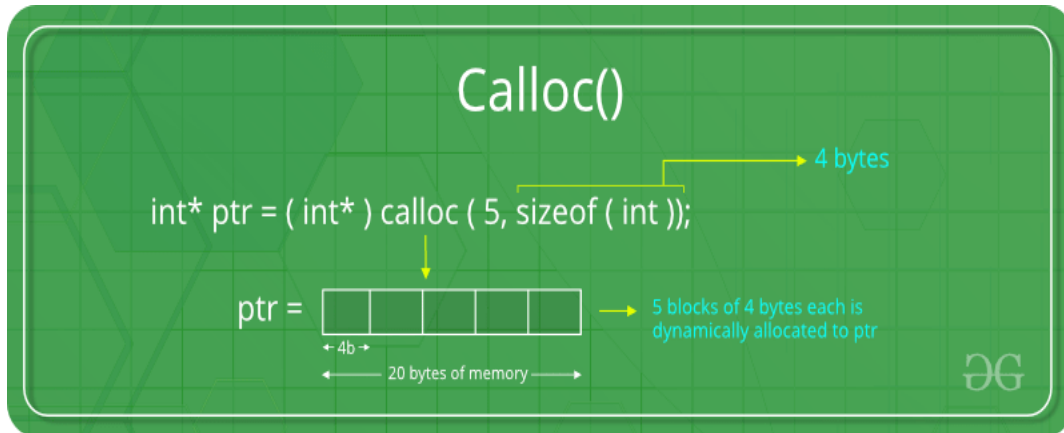
Output:

Enter number of elements: 5

Enter elements: 1, 2, 3, 4, 5

Sum is: 15

Example Program for calloc():



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

```
int main()
{
```

```
    // This pointer will hold the base address of the block created
    int* ptr;
    int n, i;
```

```
    // Get the number of elements for the array
    n = 5;
    printf("Enter number of elements: %d\n", n);
```

```
    // Dynamically allocate memory using calloc()
    ptr = (int*)calloc(n, sizeof(int));
```

```
    // Check if the memory has been successfully
    // allocated by calloc or not
    if (ptr == NULL) {
```

```
        printf("Memory not allocated.\n");
        exit(0);
```

```
    }
    else {
```

```
        // Memory has been successfully allocated
        printf("Memory successfully allocated using calloc.\n");
```

```
        // Get the elements of the array
        for (i = 0; i < n; ++i) {
            ptr[i] = i + 1;
        }
```

```
        // Print the elements of the array
```

```

        printf("The elements of the array are: ");
        for (i = 0; i < n; ++i) {
            printf("%d, ", ptr[i]);
        }

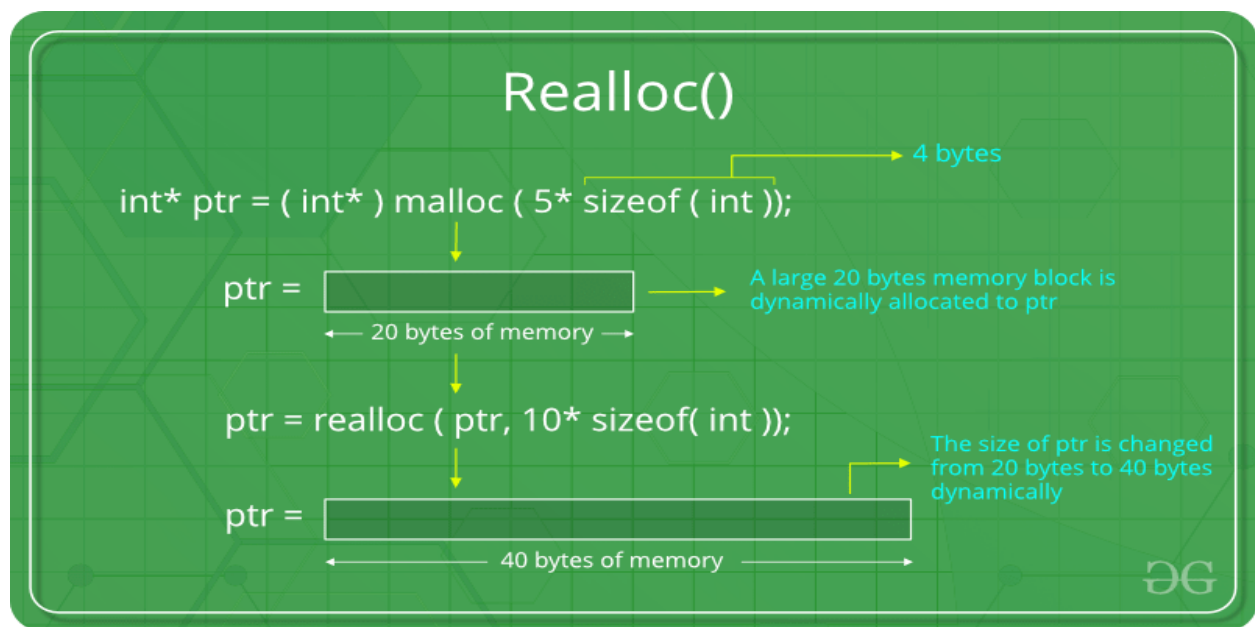
    return 0;
}

```

Output:

Enter number of elements: 5
 Memory successfully allocated using calloc.
 The elements of the array are: 1, 2, 3, 4, 5

Example Program for realloc():



```

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

```

```

int main()
{
    // This pointer will hold the base address of the block created
    int* ptr;
    int n, i;

    // Get the number of elements for the array
    n = 5;
    printf("Enter number of elements: %d\n", n);

    // Dynamically allocate memory using calloc()

```

```

ptr = (int*)calloc(n, sizeof(int));

// Check if the memory has been successfully allocated by malloc or not
if (ptr == NULL) {
    printf("Memory not allocated.\n");
    exit(0);
}
else {

    // Memory has been successfully allocated
    printf("Memory successfully allocated using calloc.\n");

    // Get the elements of the array
    for (i = 0; i < n; ++i) {
        ptr[i] = i + 1;
    }

    printf("The elements of the array are: ");
    for (i = 0; i < n; ++i) {
        printf("%d, ", ptr[i]);
    }

    n = 10;
    printf("\n\nEnter the new size of the array: %d\n", n);

    // Dynamically re-allocate memory using realloc()
    ptr = realloc(ptr, n * sizeof(int));

    // Memory has been successfully allocated
    printf("Memory successfully re-allocated using realloc.\n");

    // Get the new elements of the array
    for (i = 5; i < n; ++i) {
        ptr[i] = i + 1;
    }

    // Print the elements of the array
    printf("The elements of the array are: ");
    for (i = 0; i < n; ++i) {
        printf("%d, ", ptr[i]);
    }

    free(ptr);
}

return 0;
}

```

Output:

Enter number of elements: 5

Memory successfully allocated using calloc.

The elements of the array are: 1, 2, 3, 4, 5

Enter the new size of the array: 10

Memory successfully re-allocated using realloc.

The elements of the array are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

