

Memória dinâmica

- Até agora, criamos códigos para situações em que se conhece a quantidade exata de memória a ser usada;
- E quando não soubermos?
- Exemplo: um programa que guarda dados de N clientes de uma empresa, onde o usuário do programa decide o valor de N .

Memória dinâmica

- Solução mais simples (preguiçosa): alocar um espaço absurdo de memória.
- Resolve o problema de escassez de memória, mas também cria um novo problema.
- Solução mais indicada: MEMÓRIA DINÂMICA.

Memória dinâmica

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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```

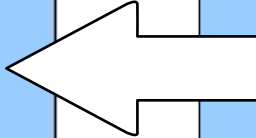
Memória dinâmica

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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```



malloc garante que
haverá um espaço
na memória de N
posições de
tamanho *char*,
liberado para
escrita e leitura

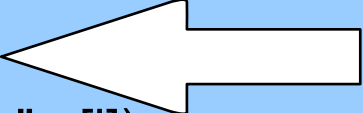
Memória dinâmica

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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```



A partir de agora,
tratamos a
variável *a* como um
vetor de 5 posições

Memória dinâmica

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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```

Este espaço na
memória está
disponível para:

ESCRITA

LEITURA

Memória dinâmica


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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```

Depois que utilizamos
esse espaço em
memória, temos de
liberá-lo
(IMPORTANTE)



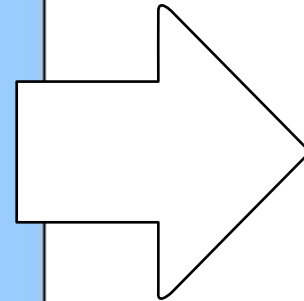
Memória dinâmica

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

void main()
{
    int N=5, i;
    char *a;

    a = (char*)malloc(N*sizeof(char));

    for(i=0;i<N;i++)
    {
        a[i] = i*2;
        printf("%d ",a[i]);
    }
    free(a);
}
```



0 2 4 6 8

Memória dinâmica

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

```
void main()
```

```
{
```

```
    int N=5, i;
```

```
    int *a;
```

```
    a = (int*)malloc(N*sizeof(int));
```

```
    for(i=0;i<N;i++)
```

```
    {
```

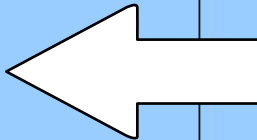
```
        a[i] = i*2;
```

```
        printf("%d ",a[i]);
```

```
    }
```

```
    free(a);
```

```
}
```



malloc garante que
haverá um espaço
na memória de N
posições de
tamanho *int*,
liberado para
escrita e leitura

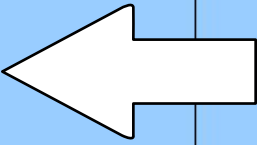
Memória dinâmica

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

void main()
{
    int N=5, i;
    float *a;

    a = (float*)malloc(N*sizeof(float));

    for(i=0;i<N;i++)
    {
        a[i] = (float)(i*2);
        printf("%d ",a[i]);
    }
    free(a);
}
```



malloc garante que
haverá um espaço
na memória de N
posições de
tamanho *float*,
liberado para
escrita e leitura

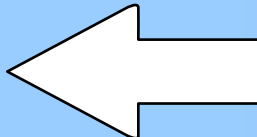
Memória dinâmica

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

void main()
{
    int N=5, i;
    float *a;

    a = (float*)malloc(N*sizeof(float));
    a[4] = 30.567;
    free(a);

    N=20;
    a = (float*)malloc(N*sizeof(float));
    a[19] = 40.537;
    free(a);
}
```



malloc reserva 5
posições de
tamanho *float*

Memória dinâmica

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

void main()
{
    int N=5, i;
    float *a;

    a = (float*)malloc(N*sizeof(float));
    a[4] = 30.567;
    free(a); ←

    N=20;
    a = (float*)malloc(N*sizeof(float));
    a[19] = 40.537;
    free(a);
}
```

A partir desse ponto
no código, não
devemos tratar
a como vetor
==> Não devemos
acessar nenhuma
posição alterada
anteriormente

Memória dinâmica


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

void main()
{
    int N=5, i;
    float *a;

    a = (float*)malloc(N*sizeof(float));
    a[4] = 30.567;
    free(a);

    N=20;
    a = (float*)malloc(N*sizeof(float));
    a[19] = 40.537;
    free(a);
}
```

Agora, podemos
tratar *a* como um
vetor de 20
posições



Memória dinâmica

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

void main()
{
    int N=5, i;
    float *a;

    a = (float*)malloc(N*sizeof(float));
    a[4] = 30.567;
    free(a);

    N=20;
    a = (float*)malloc(N*sizeof(float));
    a[19] = 40.537;
    free(a);
}
```

Finalmente,
liberamos o espaço
em memória