

// Pudio Lucas Famarane - 20.1 4003

// L03 - Alocação dinâmica de memória

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

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

```
float* reverso (int, float*);
```

```
float* alocaVetor (int);
```

```
int main (void) {
```

```
    int n;
```

```
    scanf ("%d", &n);
```

```
    float* v = alocaVetor (n);
```

```
    for (int i = 0, i < n, i++)
```

```
        scanf ("%f", &v[i]);
```

```
    float* v_inverso = reverso (n, v);
```

```
    free(v);
```

```
    for (int i = 0; i < n, i++)
```

```
        printf ("%f", v_inverso[i]);
```

```
    free(v_inverso);
```

```
    return 0;
```

```
}
```

```
float* alocaVetor (int n) {
```

```
    float* vetor = (float*) malloc (n * sizeof (float));
```

```
    if (vetor == NULL) {
```

```
        printf ("Memória insuficiente.\n");
```

```
        exit (1);
```

```
    }
```

```
    return vetor;
```

```
}
```

```
float* reverse (int n, float* v) {  
    float* v_inverse = allocVector (n);  
    for (int i = 0, i < n, i++)  
        v_inverse[i] = v[n - (i + 1)];  
    return v_inverse;  
}
```