

Functions and Program Structure

Course: Introduction to Programming and Data Structure

Laltu Sardar

Institute for Advancing Intelligence (IAI),
TCG Centres for Research and Education in Science and Technology (TCG Crest)

tcg crest

Inventing Harmonious Future

Feb 14, 2023

Dynamic Memory Allocation

- We were defining array as
`int a[N]`
- Problem: what if failed?
- What if more memory required?
- Available Function `malloc`
- Library required `stdlib.h`

Dynamic Memory Allocation

- We were defining array as
`int a[N]`
- Problem: what if failed?
- What if more memory required?
- Available Function `malloc`
- Library required `stdlib.h`

```
1  int *A ;  
2  scanf ("%d" , &N);  
3  A = (int *) malloc(N);  
4
```

Memory Allocation: `malloc`

- `malloc` allocates memory in bytes.
- Input: a positive number N
- Output: A contiguous memory of size N -bytes from RAM.
- Typecast is required.

Memory Allocation: malloc

- malloc allocates memory in bytes.
- Input: a positive number N
- Output: A contiguous memory of size N -bytes from RAM.
- Typecast is required.

Try your own

```
A = (int *) malloc(5);
```

Contiguous Allocation: calloc

```
A = (int *) calloc(N, sizeof(int));
```

- malloc just allocates memory
- calloc allocates memory and initialized with 0
- malloc is faster.

Re-allocation: realloc

```
new_ptr = (int *)realloc(old_ptr, new_size);
```

- realloc just re-allocates memory
- In general when we need to increase memory? (check what will happen if decreased)

Freeing the allocated memory

- Why? it does not automatically makes them free
- syntax:
`free(ptr);`

Swapping values of two variables

Write a function that swaps value of two integer variables.

- Take input from command line two integers a and b as `scanf("%d %d",&a,&b);`
- output the values after swapping as `printf("%d %d",a,b);`
- name the function as `swap_int()`