



VE280 Recitation Class (4)

Prepared by,
Zhang Yuhang
Yu JinZe
Cheng Songzhe
Yan Xuebin

Data Structure Summer
June 4th, 2012
F-Building, 201



Outline



- Assert
- Pointer(1)
- Function pointer
- Array
- C-string & string



Assert



- `void assert(int expression)`
 - If `expression==0` `stderr("XXX"); aborted();`
- Release Mode
 - `#define NDEBUG` // **before** including `<cassert>`
 - `g++ -DNDEBUG`
- Attention
 - One assert, one condition `//assert(i<=3&&k>=3)`
 - Can not change the environment `// assert(i++<100)`



Pointer



- syntax

```
int a=5;  
int *pa=&a;  
cout<<"Address: "<<pa<<endl;  
cout<<"Value: "<<*pa<<endl;
```

- Assign

```
int a=5, b=6;  
int *pa = &a;  
int *pb = &b;  
*pa = *pb; cout<<a<<b;  
pa = pb; cout<<a<<b;
```



Function Pointer



- syntax

```
int func(int x);
```

```
int (*f) (int x);
```

```
f = func;
```

```
a= f(a);
```

- Function pointer & “function of pointer”

```
int* f(int x);           //a function returns a pointer
```

```
int (*f) (int x);       //a pointer pointing to a function
```



Array



- `list[i]` and `*(list+i)`
- `Int a[4][3]={{1,2,3},{1,2,3},{1,2,3},{1,2,3}};`
- Passing arrays by reference to functions
`int func(int list[], unsigned int size);`
`int func(int *list, unsigned int size);`
- How to return an array?
`int[] func(int list[], unsigned int size);` ?
`int func(int list[], int newlist[], unsigned int size);`
`int * func(int list[], unsigned int size);` ?



C-string



- `char a[] = "foo";`
`char a[20]="foo";`
`char a[]={ 'f', 'o', 'o' };`
`char a[]={ 'f', 'o', 'o', '\0' };`
`const char *a="foo";`
- C-string functions
`strlen(char *s1);`
`strcpy(char *s1, char *s2);`
`strcmp(char*s1, char *s2);`



C-string



```
char a[20]="dasha";  
cout <<a <<endl;  
cout <<*a <<endl;  
cout <<*(a+1) <<endl;
```