Review programming language ‘C’

OS 4021

Erfan Zare

Fall 2023

Basic library

#include <ctype.h>

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <stdbool.h> (to use Boolean variables)

#include <conio.h>

(It is a non-standard C header file. Also it’s not supported in modern compilers such as gcc)

ctype

int isalnum(c)
Check if the passed character is alphanumeric(A-Z /1-9)
isalpha(c)
checks if the passed character is alphabetic.
isdigit(c)
checks if the passed character is decimal digit.
islower(c)
checks if the passed character is lowercase letter.
isupper(c)
checks if the passed character is uppercase letter.
tolower(c)
This function converts uppercase letters to lowercase.
toupper(c)
This function converts lowercase letters to uppercase.

stdio.h(input)

Scanf(%R",variableOfRType)
Reads formatted input from stdin. R is format specifier in

Getchar()
used to read the first character of the stdin

Gets()
Reads a line from stdin and stores it into the string pointed to by, str. It stops when either the newline character is read or when the end-of-file is reached.

getline()
This function is used to read the lines from the stdin.
i.e:
A=getline(&string,&size,stdin);

Stdio.h(output)

Printf(%R",variableOfRType)
Sends formatted output to stdout.R is format specifier in c

Putchar(a)
used to display the single character on the screen.

int puts(a)
writes a string to stdout,but not including the null character.

question

1)Compare syntax getchar vs gets
2)Compare fgets vs gets

specifier

int	%d
char	%c
float	%f
double	%lf
short int	%hd
unsigned int	%u
long int	%li
long long int	%lli
unsigned long int	%lu
unsigned long long int	%llu
signed char	%c
unsigned char	%c
long double	%Lf

Math.h

Some function:

acos,asin,atan(double x) {x in radian format.}
Cos,sin,tan(double x)
Exp,log,log10(double x)
Pow(double a,double b)
Sqrt(double x)
Floor(double x)
{largest integer value less than or equal to x}
Ceil(double x)
{smallest integer value greater than or equal to x}

question

3) What is malloc?
4)What is calloc?
Compare maaloc vs calloc

stdlib.h

Malloc(size_t size)

The “malloc” or “memory allocation” method in C is used to dynamically allocate a single large block of memory with the specified size. It returns a pointer of type void which can be cast into a pointer of any form. It doesn’t Initialize memory at execution time so that it has initialized each block with the default garbage value initially.

Example:

ptr = (int*) malloc(100 * sizeof(int));

Since the size of int is 4 bytes, this statement will allocate 400 bytes of memory. And, the pointer ptr holds the address of the first byte in the allocated memory.

calloc()

“calloc” or “contiguous allocation” method in C is used to dynamically allocate the specified number of blocks of memory of the specified type. it is very much similar to malloc() but has two different points and these are:

It initializes each block with a default value ‘0’.

It has two parameters or arguments as compare to malloc().

Example:

ptr = (float*) calloc(25, sizeof(float));

This statement allocates contiguous space in memory for 25 elements each with the size of the float.

Conditional Expressions/if statements

```c
#include <stdio.h>
int main() {
  int y;
  int x = 2;
   y = (x >= 6) ?  6 : x;
   /* This is equivalent to:  if (x >= 5)    y = 5;  else    y = x; */
   printf("y =%d ",y);
  return 0;
}
```

When use some condition can use:
|| (or) / && (and)

```c
#include<stdio.h>
int main()
{
    int marks=83;
    if(marks>75){
        printf("First class");
    }
    else if(marks>65){
        printf("Second class");
    }
    else if(marks>55){
        printf("Third class");
    }
    else{
        printf("Fourth class");
    }
    return 0;
}
```

Switch case/break

```
int day = 4;

switch (day) {
  case 6:
    printf("Today is Saturday");
    break;
  case 7:
    printf("Today is Sunday");
    break;
  default:
    printf("Looking forward to the Weekend");
}

// Outputs "Looking forward to the Weekend"
```

While/for

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

```c
int j;
j=0;
while(j<5)
{
    j++;
    printf("%d\n",j)
}
```

```c
int i;
for (i = 0; i < 5; i++) {
    printf("%d\n", i);
}
```

pointers

A pointer is an address. A pointer is a derived data type that stores a memory address. A pointer can also point to another pointer or to a function. The value of a pointer can be incremented/decremented to point to the next/previous memory location.
We can access to address of variable with use sign ‘&’ before the name of variable.
Specifier of this address is ‘%p’ that output is hex.

Declare like this:
Int *p;

Tip1:
When assign NULL to pointer,the value of that is 0.
Tip2:
&:use for return address of variable
* : 1)declare a pointer . 2) return original variables value

array

We can declare array with two ways:
1) Int num[ ]={5,1,3,2}
2) Int *num={4,2,1,6}

Also we can access to elements of array with two ways:
1) Int temp=*(num+1)
2) Int temp=num[1]

2d array

We can declare 2d array with two ways:
1) Int num[2][2]={{1,2},{4,3}} //we should write size of array

Also we can access to elements of array with two ways:
1) Int temp=*(*(num+1)+2)
2) Int temp=num[1][1]

Question:
5) Output of second code in pointers folder.

struct

Structures are a way to group several related variables into one place.
Each variable in the structure is known as a member of the structure.
Unlike an array, a structure can contain many different data types
(int, float, char, etc.).

Question:
6)Code in image below have error . why?
And How can fix it.

```c
// error? and how can fix it

struct det2 {
  int myNum;
  char myLetter;
  char myString[30];
};

int main() {
  struct det2 s1;

  s1.myString = "Some text";
  s1.myNum=2
  printf("My string: %s", s1.myString);

  return 0;
}
```

Examples

1
2
3
4
5
6
7
8
9
10
11
12
13
14

The End