Can Novice Programmers Write C Functions?

Cruz Izu & Peter Dinh

THE UNIVERSITY of ADELAIDE

Abstract

This work has analysed the solutions submitted by 301 students to a practical exam question from a CS1 course that covered C functions. We have focused on the ability of students to declare the function header correctly and to return a value to the main program.

The analysis of exam's solutions showed that most students could write a function definition with the given name and number of parameters. Most mistakes were usually related to type, either in the function return value or in the input parameter. Besides, some students were inconsistent in that they declared a correct "*int*" function but instead of returning a value, they printed either the result or a message describing the output.

It is worth noticing 16% of students developed a solution *inside* the main program instead of attempting to write a separate function, which indicates they have trouble understanding and writing modular code.

Data Collection

- We used a CS1 course to analyse the second and last question of the practical exam
  - 12 weeks of instruction covers MATLAB and C
  - 341 students enrolled – 301 attempted Q2

Exam Question

We have decided that 5 and 8 are *fortunate* digits. A positive integer is called a *fortunate* number if its decimal representation has only fortunate digits, i.e. 5, 55, 585

Write a function in C called ***isFortunate*** that receives an integer number as a parameter, and returns

- 1 if the number if fortunate,
- 0 if is the number is not fortunate
- -1 if the number is negative.

Data Analysis Methodology

Breakdown the question into its components

1. Function header + return statement
2. Extract/split digits
3. Check fortunate conditions

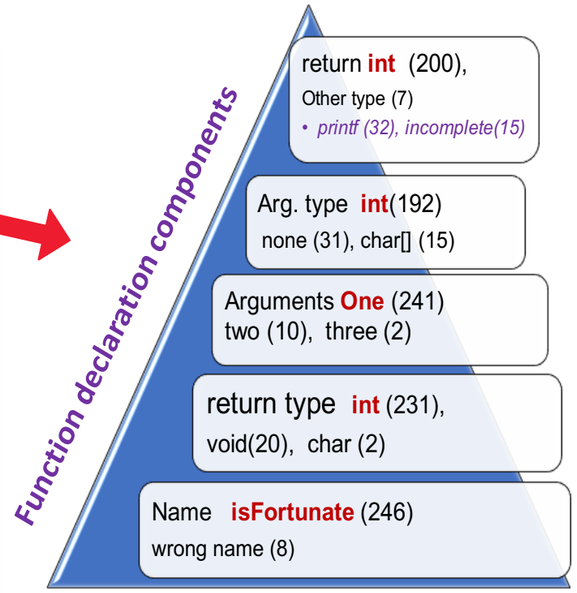
Iteration 1

Header type
Header name
Argument type/s
Return type

Type of iteration used
Digit extraction

Iterations 2-3

An **iterative** process was used to develop a criteria that can clearly identify student coding **strategies**




```c
int isFortunate(int n)
{
  int i=0,j =0;      //variables assigned
  int rem, temp;
  temp=n;          //creates a template

  if (temp<0)
  {
    return -1;
  }else{
    while (temp!=0)
    {
      i++;       //counts the number of digits
      rem=temp%10;     //gets the last digit
      if (rem==3||rem==7) //is the digit a 3 or 7?
      {
        j++;
      }
    temp=temp/10;      //moves to next digit
    }
  }
  if (j==i)      // all fortunate
  {
    return 1;
  }else{
    return 0;
  }
}
```

Modulo + division

```c
int isFortunate(x)
{
  int a = (x/100);
  int b = ((x-100*a)/10);    //determine the value of the
  int c = (x-100*a-10*b);    //'hundreds', 'tens' and 'ones'

  //determines whether all of the numbers are either 5 or 8
  if((a==5 || a==8 || a == 0) && (b==5 || b==8 || b ==0)
      && (c==5 || c==8 || c ==0))
    return(1);
  else
    return(0);
}
```

fixed length digit extraction

```c
int isFortunate(int A[i])
{
  int lenA;
int i;

for(i=0;i<lenA;i++)
{
if (A[i]==3||A[i]==7)
  printf("%d is fortunate",A[i]);
else
  printf("%d is not fortunate",A[i]);
}
}
```

≠ Array indexing

```c
void isFortunate(int a)
{

 if (a==37 || a==73 || a==3 || a==7 || a==373 || a==377 || a==733 || a==773 || a==737)
 {
 printf("1");
 }
  if (a < 0 )
  {
  printf("-1");
  }
 if (a!=37 && a!=73 && a!=3 && a!=7 && a!=373 && a!=377 && a!=733 && a!=773 && a!=737 && a >0)
 {
 printf("0");
 }
}
```

enumeration

Conclusions

- 66% of novice programmers that wrote a function header did it correctly. Most frequent mistakes due to missing or incorrect **type** errors. However, 50 students (16%) wrote the task inside main, showing a clear lack of understanding.
- **Consistency** in declared and returned types is sometimes ignored - should be stressed in teaching activities
- Students used partially correct strategies to extract/split digit part. Some strategies account for wrong function header
- Overall achievements for novice programmers in line with other programming topics.

ACSW 2019 (previously presented at TALE 2018 as short paper)

Sydney, January 2019