Fpt University

Slot 12
Programming With Menu

A review for C-Functions

Pointers are parameters of functions

Why is Menu?

- Generally, a program performs some operations and at a time only one task is carried out. ➔ A menu is usually used.

➔ How are menus implemented in C program?

Idea

- Common Algorithm:

```
int userChoice;
do
{  userChoice= getUserChoice();
   switch (userChoice)
   {  case 1: function1(); break;
      case 2: function2(); break;
      case 3: function3(); break;
   }
}
while (userChoice >0 && userChoice<maxChoice);
```

Problem

- Write a C program using the following menu:

  1- Operation 1

  2- Operation 2

  Others- Quit
- If user chooses 1, user will input 2 integers, the program will print out sum of integers between them including them.
- If user chooses 2, user will input 2 characters, the program will print out the ASCII table between two inputted characters in ascending order.
- If user chooses other options, the program will terminate.

Implementation: menuDemo1.c

```c
/* MenuDemo1.c */
#include <stdio.h>
/* Function for getting a choice from user - Menu */
int getUserChoice()
{ int choice; /* choice from user */
  /* print out the meu */
  printf("\n1-Operation 1");
  printf("\n2-Operation 2");
  printf("\nOthers-Quit");
  /* Accept user choice */
  printf("\nChoose:");
  /* %*c: Remove the ENTER key but no character variable
  is needed*/
  scanf("%d%*c", &choice);
  return choice;
}
```

Implementation: menuDemo1.c

```
17 /* Function for operation 1
18    user will input 2 integers, the program will print out
19    sum of integers between them including them
20 */
21 int sumBetween(int a, int b)
22 {   int t;
23     if (a>b) /* a must be less than
24     {  t= a; a=b; b=t;
25     }
26     int S= 0;
27     for (t=a; t<=b; t++) S+=t;
28     return S;
29 }
30 void function1()
31 {   int n1, n2; /* 2 integers */
32     printf("Enter 2 integers:");
33     scanf("%d%d%*c", &n1, &n2);
34     printf("Sum=%d\n",sumBetween(n1, n2));
35 }
```

```
1-Operation 1
2-Operation 2
Others-Quit
Choose:1
Enter 2 integers:9 5
Sum=35
```

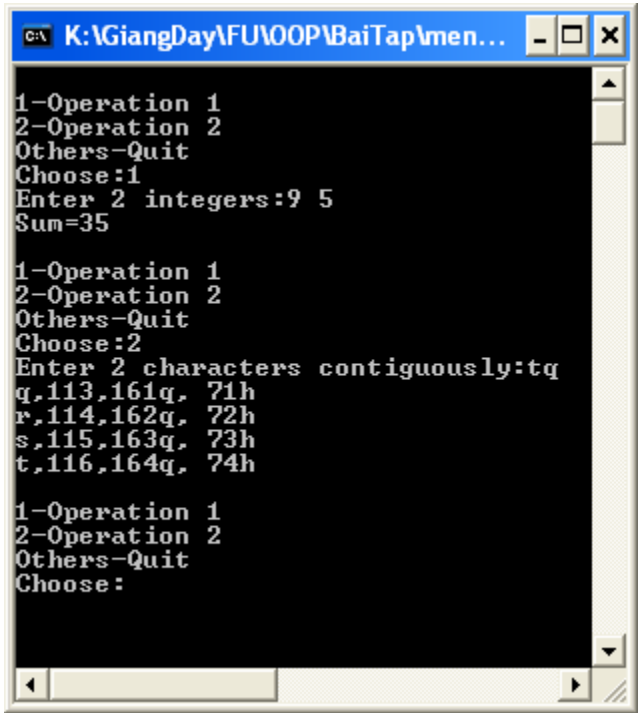
Implementation: menuDemo1.c

```
36 /*  Operation 2:user will input 2 characters, the program
37     will print out the ASCII table between two inputted
38     characters in ascending order.
39 */
40 /* Print ASCII table betwwen 2 characters,ascending order
41 void printAscii(char c1, char c2)
42 {   char c;
43     if (c1>c2) /* c1 must be less than c2 */
44     {   c=c1; c1=c2; c2=c;
45     }
46     for (c=c1; c<=c2; c++)
47       printf("%c,%3d,%3oq,%3Xh\n", c,c,c,c);
48 }
49 void function2()
50 {   char c1, c2; /* inputted characters */
51     printf("Enter 2 characters contiguously:");
52     scanf("%c%c", &c1, &c2);
53     printAscii(c1, c2);
54 }
```

```
1-Operation 1
2-Operation 2
Others-Quit
Choose:2
Enter 2 characters contiguously:tq
q,113,161q, 71h
r,114,162q, 72h
s,115,163q, 73h
t,116,164q, 74h
```

Implementation: menuDemo1.c

```
55
56 int main()
57 {    int userChoice;
58      do
59      {   userChoice= getUserChoice();
60          switch(userChoice)
61          {   case 1: function1(); break;
62              case 2: function2(); break;
63              default: printf("Bye!\n");
64          }
65      }
66      while (userChoice>0 && userChoice<3);
67      fflush(stdin);
68      getchar();
69      return 0;
70 }
```



Functions with pointers as parameters

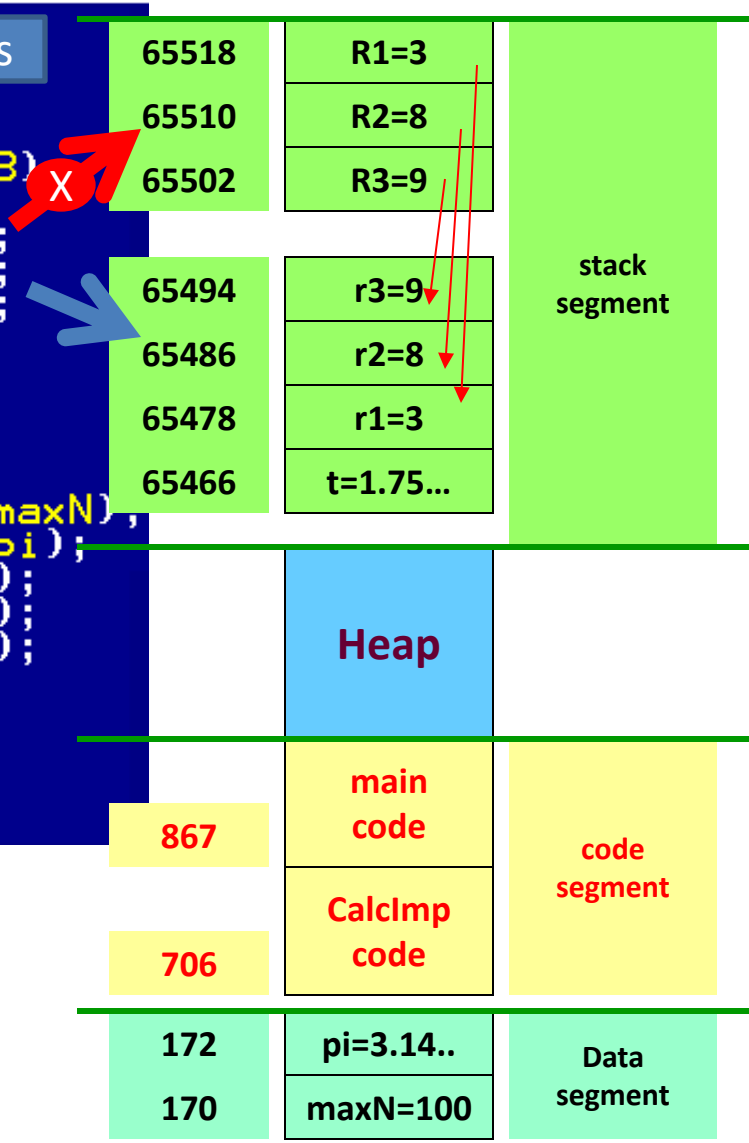
- C uses by-value parameters only → A function can not modify values of arguments.
- To modify values of arguments, pointers as parameters of a function are used.

Functions with pointers as parameters

Review: Pass by-value Parameters

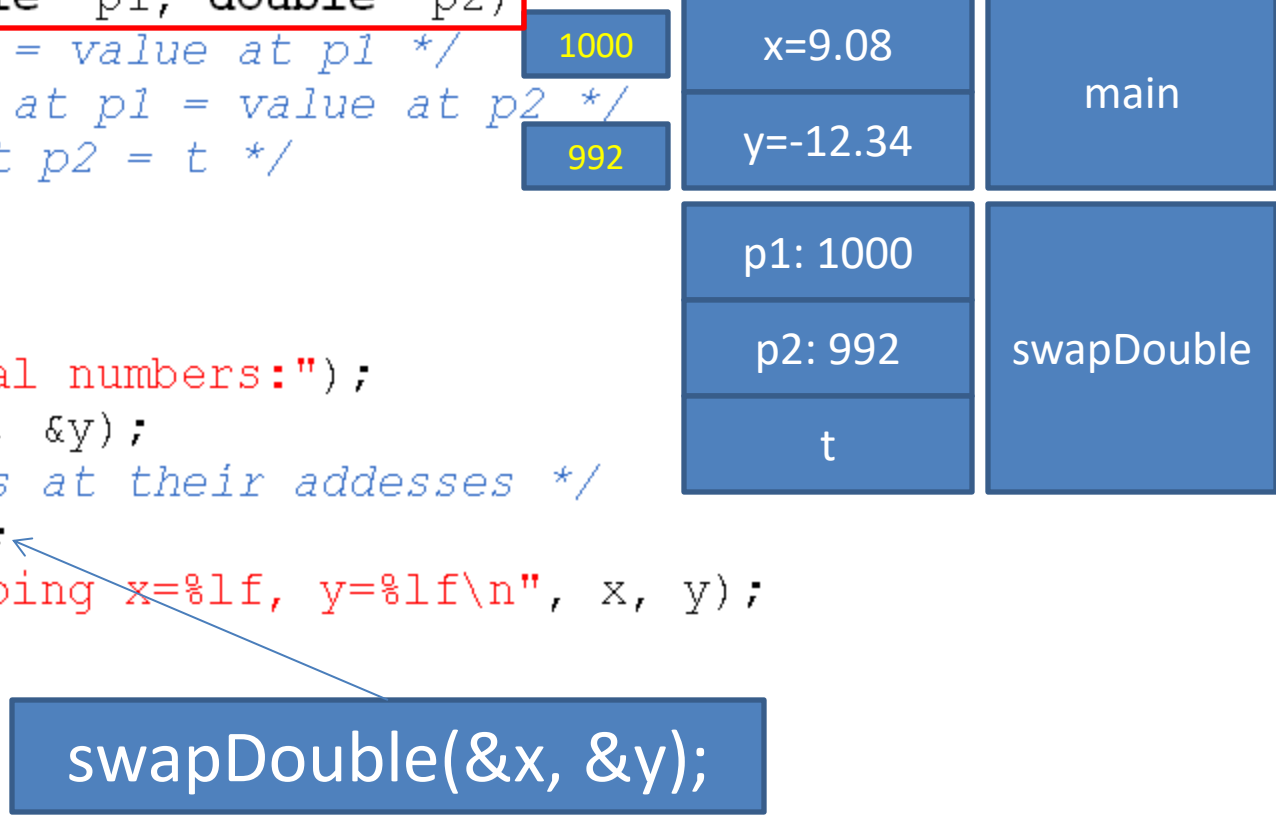
```
#include<stdio.h>
#include <conio.h>
int maxN=100;
double pi=3.141592;
double CalcImp(double r1, double r2, double r3)
{   double t=1/(1/r1 + 1/r2 + 1/r3);
    printf("r1    Addr:%u,value:%lf\n", &r1,r1);
    printf("r2    Addr:%u,value:%lf\n", &r2,r2);
    printf("r3    Addr:%u,value:%lf\n", &r3,r3);
    printf("t     Addr:%u,value:%lf\n", &t ,t);
   return t;
}
int  main()
{ double R1=3, R2=8, R3=9;
   printf("maxN Addr:%u,  value:%d \n", &maxN,maxN);
   printf("pi   Addr:%u,  value:%lf\n", &pi  ,pi);
   printf("R1   Addr:%u,value:%lf\n", &R1  ,R1);
   printf("R2   Addr:%u,value:%lf\n", &R2  ,R2);
   printf("R3   Addr:%u,value:%lf\n", &R3  ,R3);
   printf("main    addr:%u\n", &main);
   printf("CalcImp addr:%u\n", &CalcImp);
   printf("Impedance: %lf",CalcImp(R1,R2,R3));
   getch();
}
```

```
maxN Addr:170,  value:100
pi   Addr:172,  value:3.141592
R1   Addr:65518,value:3.000000
R2   Addr:65510,value:8.000000
R3   Addr:65502,value:9.000000
main    addr:867
CalcImp addr:706
r1   Addr:65478,value:3.000000
r2   Addr:65486,value:8.000000
r3   Addr:65494,value:9.000000
t    Addr:65466,value:1.756098
Impedance: 1.756098
```



Pointers as parameters: Demo

```
1  /* Accept 2 numbers, swap them, then print out them */
2  #include <stdio.h>
3  /* SWAPPING 2 DOUBLE NUMBERS AT ADDRESSES p1, p2 */
4  void swapDouble (double *p1, double *p2)
5  {   double t=*p1; /* t = value at p1 */
6      *p1= *p2; /* value at p1 = value at p2 */
7      *p2= t; /* value at p2 = t */
8  }
9  int main()
10 {   double x, y;
11     printf("Enter 2 real numbers:");
12     scanf("%lf%lf", &x, &y);
13     /* swaping 2 values at their addesses */
14     swapDouble2 (x, y);
15     printf("After swapping x=%lf, y=%lf\n", x, y);
16     fflush(stdin);
17     getchar();
18     return 0;
19 }
```



Pointers as parameters: Demo

```
9  void swapDouble2 (double p1, double p2)
10 {   double t=p1;
11     p1= p2;
12     p2= t;
13 }
14
15 int main()
16 {   double x, y;
17     printf("Enter 2 real numbers:");
18     scanf("%lf%lf", &x, &y);
19     /* swaping 2 values at their addesses */
20     swapDouble2 (x, y);
21     printf("After swapping x=%lf, y=%lf\n", x, y);
22     fflush(stdin);
23     getchar();
24     return 0;
25 }
```