Let Us C

Manish

December 6, 2023

Chapter 3

The Loop Control Structure

while loop

[A] What would be the output of the following programs:

a j is not initialized with any value so it will use the garbageValue already present in it. Making the output uncertain.

b

```
1
2
3
4
5
6
7
8
9
10
```

c same as part a of this section.

d

```
0
```

e

```
0
```

f syntax error in while statement.

g

```
2 3 3
```

h

```
3 3 1
```

i

```
malyalam is a palindrome
malyalam is a palindrome
(... infinite loop)
```

j

```
A computer buff!
A computer buff!
(... infinite loop)
```

k

```
10
10
(... infinite loop)
```

l

```
1.100000
```

m

```
In while loop
In while loop
(... infinite loop)
```

n

```
Ascii value 0 Character
Ascii value 1 Character
...
...
Ascii value 127 Character
Ascii value -128 Character
Ascii value -127 Character
...
...
```

```
Ascii value -1 Character
Ascii value 0 Character
Ascii value 1 Character
...
(... infinite loop)
```

o

```
3 1
1 3
0 4
-1 5
```

p

```
4 0
3 1
```

[B] Attempt the following:

a

```
int main() {
  int hWorked = 0, overtime = 0;
  printf("enter number of hours worked by employees: ");
  scanf("%d", &hWorked);
  overtime = (hWorked > 40)? 12*(hWorked-40): 0;

  printf("overtime pay = %d\n", overtime);
  return 0;
}
```

b

```
int main() {
  int num = 0, fact = 1;
  printf("enter number: ");
  scanf("%d", &num);
  while(num > 0) {
    fact = fact*num;
    num--;
  }
  printf("factorial = %d\n", fact);
  return 0;
}
```

c

```
int main() {
  int num1 = 0, num2 = 0, res = 1;
  printf("enter 2 numbers: ");
```

```
  scanf("%d%d", &num1, &num2);
  while(num2 > 0) {
    res *= num1;
    num2--;
  }
  printf("num1 raised to the num2 = %d\n", res);
  return 0;
}
```

d

```
int main() {
  int x = 0;
  while(x <= 255) {
    printf("Ascii value %d = %c\n", x, x);
    x++;
  }
  return 0;
}
```

e

```
int main() {
  int num = 1, d1, d2, d3;
  while(num <= 500) {
    //the three digits of number 544 = d1d2d3
    d1 = num/100;
    d2 = num/10 % 10;
    d3 = num%10;
    if (d1*d1*d1 + d2*d2*d2 + d3*d3*d3 == num) {
      printf("%d ", num);
    }
    num++;
  }
  return 0;
}
```

f

```
int main() {
  int matchsticks = 21, user;
  int r = 1;
  while(matchsticks > 0) {
    printf("\nRound %d\n", r++);
    printf("your move: \t\t");
    scanf("%d", &user);
    matchsticks -= user;
    if(matchsticks <= 0) {
      printf("\n\nremaining matchsticks = 0\nYOU LOSE!!!\n\n");
      break;
```

```
    }
    printf("my move: \t\t%d", 5-user);
    matchsticks -= 5-user;
    printf("\nremaining matchsticks = %d\n", matchsticks);
  }
  return 0;
}
```

g

```
int main() {
  int num, pve, nve, zs, choice;
  pve = nve = zs = 0;
  choice = 1;
  while(choice == 1) {
    printf("\nyour number: ");
    scanf("%d", &num);
    if(num > 0) {
      pve++;
    } else if(num < 0) {
      nve++;
    } else {
      zs++;
    }
    printf("continue?(1/0)\t");
    scanf("%d", &choice);
  }
  printf("numbers entered: \n+ve = %d\n-ve = %d\n0 = %d\n", pve, nve, zs);
  return 0;
}
```

h

```
int main() {
  int num, oct = 0, digits = 1;
  printf("\nyour number: ");
  scanf("%d", &num);
  printf("octal equivalent of %d = ", num);
  while(num > 0) {
    oct = ((num%8) * digits) + oct;
    num /= 8;
    digits *= 10;
  }
  printf("%d\n", oct);
  return 0;
}
```

i

```
 int main() {
```

```
  int num, min, max, choice;
  printf("\nyour number: ");
  scanf("%d", &num);
  min = max = num;
  printf("\ncontinue? (1/0) ");
  scanf("%d", &choice);
  while(choice == 1) {
    printf("\nyour number: ");
    scanf("%d", &num);
    if(min > num) {
      min = num;
    } else if(max < num) {
      max = num;
    }
    printf("\ncontinue? (1/0) ");
    scanf("%d", &choice);
  }
  printf("range of entered numbers = %d\n", max-min);
  return 0;
}
```

for, break, continue, do-while

[C] What would be the output of the following programs:

a no output

b

```
2
3
4
5
6
```

c

```
2
5
```

d

```
A
A
A
A
A
```

[D] Answer the following:

a initialize loop counter
test
incrementing/decrementing counter

b arithmetic, relational, assignment

c a for loop

d at least once

e initialization, execution of body, testing

f 3 is not an infinite loop

g continue

[E] Attempt the following:

a
```
#include <stdio.h>
#include <math.h>

int main() {
  int num, i, sr, isPrime;
  for(num = 1; num <= 300; num++) {
    isPrime = 1;
    i = 2;
    sr = sqrt(num);
    for(i = 2; i <= sr; i++) {
      if(num%i == 0) {
        isPrime = 0;
        break;
      }
    }
    if(isPrime) {
      printf("%d ", num);
    }
  }
  return 0;
}
```

b
```
int main() {
  int i;
  for(i = 1; i <= 30000; i++) {
    printf("%c", 1);
    \\ printf("%s", "\u263a");
  }
  return 0;
}
```

c
```
int main() {
  int i, num;
  float sum = 0, fact;
  for(i = 1; i <= 7; i++) {
    num = i;
    fact = 1;
    while(num > 0) {
      fact = fact*num;
      num--;
    }
    sum += i/fact;
  }
  printf("sum of first 7 terms = %f\n", sum);
  return 0;
}
```

d
```
int main() {
  int i, j, k;
  for(i = 1; i <= 3; i++) {
    for(j = 1; j <= 3; j++) {
      for(k = 1; k <= 3; k++) {
        printf("%d%d%d\n", i, j, k);
      }
    }
  }
  return 0;
}
```

e
```
int main() {
  float i, x;
  int y;
  printf("\t\t\t1\t\t\t2\t\t\t3\t\t\t4\t\t\t5\t\t\t6");
  for(x = 5.5; x <= 12.5; x += 0.5) {
    printf("\n%4.1f\t", x);
    for(y = 1; y <= 6; y++) {
```

```
      i = 2 + (y + (0.5 * x));
      printf("%.2f\t", i);
    }
  }
  return 0;
}
```

f
```
int main() {
  int i, j;
  for(i = 0; i < 7; i++) {
    for(j = 0; j < 13; j++) {
      if((7-i <= j && j < 7) || (i-1 > j%7 && j/7)) {
        printf("  ");
        continue;
      }
      if(j < 7) {
        printf("%c ", 'A' + j);
      } else {
        printf("%c ", 'F' - j%7);
      }
    }
    printf("\n");
  }
  return 0;
}
```

g
```
int main() {
  int i;
  for(i = 0; i < 30000; i++) {
    //printf("%c%c", 3, 4);
    printf("%s%s", "\u2661", "\u2662");
  }
  return 0;
}
```

h
```
int main() {
  int i, num;
  scanf("%d", &num);
  for(i = 1; i <= 10; i++) {
    printf("%d * %2d = %3d\n", num, i, num*i);
  }
  return 0;
}
```

i
```
int main() {
  int i, j, num = 0;
  for(i = 0; i < 4; i++) {
    for(j = 0; j < 4; j++) {
      if(i+j < 3) {
        printf(" ");
      } else {
        printf("%d ", ++num);
      }
    }
    printf("\n");
  }
  return 0;
}
```

j
```
int main() {
  int i, j, n, nmr, r, nfact, nmrfact, rfact;
  for(i = 0; i < 5; i++) {
    for(j = 5-i; j > 0; j--) {
      printf(" ");
    }
    for(j = 0; j <= i; j++) {
        n = i; r = j; nmr = n-r;
        nfact = nmrfact = rfact = 1;
        while(n > 0) {
          nfact *= n;
          n--;
        }
        while(nmr > 0) {
          nmrfact *= nmr;
          nmr--;
        }
        while(r > 0) {
          rfact *= r;
          r--;
        }
        printf("%d ", nfact / (nmrfact*rfact));
    }
    printf("\n");
  }
  return 0;
}
```

k
```
int main() {
  int mCost = 6000, mEarning = 1000, mSalvage = 2000;
```

```
  int year = 0;
  float altEarn = 0, mEarn = 0, iRate = 12;
  while(altEarn >= mEarn) {
    altEarn += (altEarn + 4000) * iRate / 100;
    mEarn += 1000;
    year++;
  }
  printf("minimum life = %d\n", year);

  return 0;
}
```

l

```
int main() {
  float p, r, n, q, nq;
  double amount = 0, exp, expNq;
  int i = 0, j;
  while(i < 10) {
    scanf("%f%f%f%f", &p, &r, &n, &q);
    j = 0;
    exp = 1 + r/q;
    expNq = 1;
    nq = n*q;
    while(j < nq) {
      expNq *= exp;
      j++;
    }
    amount = p * expNq;
    printf("amount = %lf\n", amount);
    i++;
  }

  return 0;
}
```

m

```
int main() {
  float x;
  scanf("%f", &x);
  float exp, expn, nlog = (x-1)/x;
  expn =  exp = nlog;

  int i = 2;
  while(i < 8) {
    expn *= exp;
    nlog += expn/2;
    i++;
```

```
  }
  printf("natural log of first terms = %f\n", nlog);

  return 0;
}
```

Chapter 4

The Case Control Structure

[A] What would be the output of the following programs:

a

```
Heart
I thought one wears a suite
```

b

```
I am in case 3
```

c

```
Pure Simple Egghead!
```

d

```
Customers are dicey
Markets are pricey
Inverstors are moody
At least employees are good
```

e

```
Trapped
```

f

```
You entered a and b
```

g

```
Feeding fish
Weeking grass
mending roof
```

```
Just to survive
```

[B] Point out the errors, if any, in the following programs:

a
```
syntax error in case 0 & 1;
Also case statements are not allowed outside switch statement.
```

b
```
error: expression in case is not integer constant. (operand is not a constant)
```

c
```
error: quantity in switch is not an integer.
```

d
```
error: 2nd case statement is not an integer constant,
variables a and b are not considered constatns.
```

[C] Write a menu driven program which has following options:

1. Factorial of a number.
2. Prime or not
3. Odd or even
4. Exit

```
#include <stdio.h>
#include <math.h>

int main() {
  int choice, i, sr, num, fact;
  while(1) {
    printf("\n1. Factorial");
    printf("\n2. Prime");
    printf("\n3. Odd/Even");
    printf("\n4. Exit");
    printf("\nYour choice?");
    scanf("%d", &choice);

    switch (choice) {
      case 1:
        printf("\nenter number: ");
        scanf("%d", &num);
        fact = 1;
        while(num > 0) {
          fact = fact*num;
```

```
            num--;
          }
          printf("factorial = %d\n", fact);
          break;
        case 2:
          printf("\nenter number: ");
          scanf("%d", &num);
          i = 2;
          sr = sqrt(num);
          for(i = 2; i <= sr; i++) {
            if(num%i == 0) {
              printf("%d is a prime number.\n", num);
              break;
            }
          }
          break;
        case 3:
          printf("\nenter number: ");
          scanf("%d", &num);
          if(num%2 == 0) {
            printf("%d is an even number.\n", num);
          } else {
            printf("%d is an odd number.\n", num);
          }
          break;
        case 4:
          return 0;
      }
    }

    return 0;
  }
```

[D] Write a program to find the grace marks for a student using switch. The user should enter the class obtained by the student and the number of subjects he has failed in.:

```
#include <stdio.h>

int main() {
  int class, noOfSubs, grace = 0;
  printf("enter class obtained by student: ");
  scanf("%d", &class);
  printf("number of subjects failed in: ");
  scanf("%d", &noOfSubs);
```

```
  switch (class) {
    case 1:
      if(noOfSubs <= 3) {
        grace += noOfSubs * 5;
      }
      break;
    case 2:
      if(noOfSubs <= 2) {
        grace += noOfSubs * 4;
      }
      break;
    case 3:
      if(noOfSubs <= 1) {
        grace += noOfSubs * 5;
      }
      break;
  }

  printf("grace marks for student = %d\n", grace);

  return 0;
}
```