CSE notes for final

If you find any correction, please contact: Akib Reza 1061

Switch:

switch.c | parameterArgument.c

```c
#include <stdio.h>

//switch statement

int main()
{
    int x,y,result;

    printf("Enter two numbers: ");
    scanf("%d%d",&x,&y);

    int choice;
    printf("1 = Add \n2 = Subtract \n");
    printf("Choose add or subtract: ");
    scanf("%d",&choice);

    switch(choice)
    {
    case 1:
        result = x + y;
        break;
    case 2:
        result = x - y;
        break;
    }
    printf("The result is: %d",result);

    return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\switch.exe"

```
Enter two numbers: 4 8
1 = Add
2 = Subtract
Choose add or subtract: 1
The result is: 12
Process returned 0 (0x0)   execution time : 4.663 s
Press any key to continue.
```

ascendingTriangle.c | break.c | callByReference.c | callByValue.c | continue.c | descendingTriangle.c | doWhileLoop.c | forLoop.c | goto.c | maxMinArray.c | parameterArgument.c | recursionFactorial.

```c
#include <stdio.h>

//break statement

int main() {
  int i;
  for (i = 1; i <= 10; i++) {
    if (i == 5) {
      break;
    }
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\break.exe"

```
1 2 3 4
Process returned 0 (0x0)   execution time :
0.044 s
Press any key to continue.
```

ascendingTriangle.c | break.c | callByReference.c | callByValue.c | continue.c | descendingTriangle.c | doWhileLoop.c | forLoop.c | goto.c | maxMinArray.c | parame

```c
#include <stdio.h>

//continue statement

int main() {
  int i;
  for (i = 1; i <= 10; i++) {
    if (i % 2 != 0) {
      continue;
    }
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final...

```
2 4 6 8 10
Process returned 0 (0x0)   execution tim
e : 0.045 s
Press any key to continue.
```

ascendingTriangle.c | break.c | callByReference.c | callByValue.c | continue.c | descendingTriangle.c | doWhileLoop.c | forLoop.c | goto.c | maxMinArray.c | parameterArgument.c | recursio

```c
#include <stdio.h>

//goto statement

int main()
{
  printf("1\n");
  printf("2\n");

  goto jumpLine;

  printf("3\n");
  printf("4\n");

  jumpLine:
  printf("10");

  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\goto.exe"

```
1
2
10
Process returned 0 (0x0)   execution time : 0.044 s
Press any key to continue.
```

<u>The main difference between while, do while and for loop is:</u>

<u>While loop:</u> The loop body is executed as long as the condition is true. If the condition is false initially, the loop body is never executed.

<u>Do while loop:</u> The loop body is executed at least once, and then the condition is checked. If the condition is true, the loop body is executed again.

<u>For loop:</u> It consists of a loop variable initialization, a termination condition, and an increment/decrement. The loop body is executed as long as the termination condition is true.

Example:

whileLoop.c | doWhileLoop.c | forLoop.c

```c
#include <stdio.h>

//while loop:

int main() {
  int i = 1;
  while (i <= 10) {
    printf("%d ", i);
    i++;
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignment...

```
1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution
 time : 0.044 s
Press any key to continue.
```

whileLoop.c | doWhileLoop.c | forLoop.c

```c
#include <stdio.h>

//do while loop:

int main() {
  int i = 1;
  do {
    printf("%d ", i);
    i++;
  } while (i <= 10);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\do...

```
1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution time
: 0.040 s
Press any key to continue.
```

whileLoop.c | doWhileLoop.c | forLoop.c

```c
#include <stdio.h>

//for loop:

int main() {
  int i;
  for (i = 1; i <= 10; i++) {
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\fo...

```
1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution time
: 0.042 s
Press any key to continue.
```

Triangle Pattern:

ascendingTriangle.c descendingTriangle.c

```c
#include <stdio.h>

//*
//* *
//* * *
//* * * *
//* * * * *

int main() {
  int i, j, rows;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
  for (i = 1; i <= rows; i++) {
    for (j = 1; j <= i; j++) {
      printf("* ");
    }
    printf("\n");
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\ascend...

```
Enter the number of rows: 6
*
* *
* * *
* * * *
* * * * *
* * * * * *

Process returned 0 (0x0)   execution time : 2.238 s

Press any key to continue.
```

ascendingTriangle.c descendingTriangle.c

```c
#include <stdio.h>

//* * * * *
//* * * *
//* * *
//* *
//*

int main() {
  int i, j, rows;
  printf("Enter the number of rows: ");
  scanf("%d", &rows);
  for (i = 1; i <= rows; i++) {
    for (j = rows; j >= i; j--) {
      printf("* ");
    }
    printf("\n");
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\C...

```
Enter the number of rows: 6
* * * * * *
* * * * *
* * * *
* * *
* *
*

Process returned 0 (0x0)   executio
n time : 0.727 s
Press any key to continue.
```

Call by value and Call by reference:

Call by value: In this method, a copy of the actual argument is passed to the formal parameter. Changes made to the formal parameter don't affect the actual argument.

Call by reference: In this method, a reference (memory address) to the actual argument is passed to the formal parameter. Changes made to the formal parameter directly affect the actual argument.

callByReference.c | callByValue.c | continue.c | maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | series1.c | series2.c | series3.c | stringLength.c

```c
#include <stdio.h>

//call by value

void swap(int a, int b)
{
    int temp = a;
    a = b;
    b = temp;
}

int main()
{
    int x = 10, y = 20;
    swap(x, y);
    printf("x = %d, y = %d", x, y);
    return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\callByValue.exe"

```
x = 10, y = 20
Process returned 0 (0x0)   execution time : 0.0
54 s
Press any key to continue.
```

callByReference.c | continue.c | maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | series1.c | series2.c | series3.c | stringLength.c | stringRev

```c
#include <stdio.h>

//call by reference

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int x = 10, y = 20;
    swap(&x, &y);
    printf("x = %d, y = %d", x, y);
    return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\callByRefere...

```
x = 20, y = 10
Process returned 0 (0x0)   execution tim
e : 0.041 s
Press any key to continue.
```

Series print with for loop examples:

maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | series1.c | series2.c | series3.c | stringLength.c | stringReverse.c | structure.c | sum.c

```c
#include <stdio.h>

// Use loop to write this series: 1, 2 ,3 ,4, 5

int main() {
  int i;
  for (i = 1; i <= 5; i++) {
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse...

```
1 2 3 4 5
Process returned 0 (0x0)   execution
time : 0.044 s
Press any key to continue.
```

maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | series2.c | series3.c | stringLength.c | stringReverse.c | structure.c | sum.c | switch.c | to

```c
#include <stdio.h>

// Use loop to write this series: 2,4,6,8,10

int main() {
  int i;
  for (i = 2; i <= 10; i = i+2) {
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignm...

```
2 4 6 8 10
Process returned 0 (0x0)   execution t
ime : 0.045 s
Press any key to continue.
```

maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | series3.c | stringLength.c | stringReverse.c | structure.c | sum.c | switch.c | toggleCase.c

```c
#include <stdio.h>

// Use loop to write this series: 2,4,8,16,32

int main() {
  int i;
  for (i = 2; i <= 32; i = i*2) {
    printf("%d ", i);
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\...

```
2 4 8 16 32
Process returned 0 (0x0)   execution tim
e : 0.045 s
Press any key to continue.
```

Array:

maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | stringLength.c | stringReverse.c | structure.c | sum.c | switch.c | toggleCase.c

```c
#include <stdio.h>
#include <string.h>

//toggle case in array

int main() {
  char str[100];
  int i;
  printf("Enter a string: ");
  scanf("%s", str);
  for (i = 0; i < strlen(str); i++) {
   if (str[i] >= 'a' && str[i] <= 'z') {
     str[i] = str[i] - 32;
   } else if (str[i] >= 'A' && str[i] <= 'Z') {
     str[i] = str[i] + 32;
   }
  }
  printf("Toggled string: %s\n", str);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\toggleCase.e...

```
Enter a string: ErLiNGhaaLAND
Toggled string: eRlIngHAAland

Process returned 0 (0x0)   execution time : 14.154 s
Press any key to continue.
```

maxMinArray.c | parameterArgument.c | recursionFactorial.c | recursionFibonacci.c | searching.c | stringLength.c | stringReverse.c | structure.c | sum.c | switch.c

```c
#include <stdio.h>
#include <string.h>

//character search in array

int main() {
  char str[100], x;
  int i, flag = 0;
  printf("Enter a string: ");
  scanf("%s", str);
  printf("Enter the character to be searched: ");
  scanf(" %c", &x);
  for (i = 0; i < strlen(str); i++) {
   if (str[i] == x) {
    flag = 1;
    break;
   }
  }
  if (flag == 1) {
   printf("Character found at index %d\n", i);
  } else {
   printf("Character not found\n");
  }
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\searching.exe"

```
Enter a string: mahrezmusa
Enter the character to be searched: z
Character found at index 5

Process returned 0 (0x0)   execution time : 16.453 s
Press any key to continue.
```

recursionFactorial.c | recursionFibonacci.c | stringLength.c | stringReverse.c | structure.c | switch.c | parameterArgument.c | maxMinArray.c | *sum.c

```c
#include <stdio.h>

//summation and average of numbers in array

int main() {
  int n, i, sum = 0;
  printf("Enter the number of elements in the array: ");
  scanf("%d", &n);

  int a[n];
  printf("Enter the elements of the array: ");

  for (i = 0; i < n; i++) {
    scanf("%d", &a[i]);
  }

  for (i = 0; i < n; i++) {
    sum = sum + a[i];
  }

  printf("Sum of the elements in the array: %d\n", sum);

  float average = (float)sum / n;
  printf("Average: %.2f",average);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\sum.exe"

```
Enter the number of elements in the array: 3
Enter the elements of the array: 4 2 7
Sum of the elements in the array: 13
Average: 4.33
Process returned 0 (0x0)   execution time : 4.672 s
Press any key to continue.
```

recursionFactorial.c | recursionFibonacci.c | stringLength.c | stringReverse.c | structure.c | switch.c | parameterArgument.c | maxMinArray.c

```c
#include <stdio.h>

//maximum and minimum of array

int main() {
  int n, i;
  int max, min;

  printf("Enter the number of elements in the array: ");
  scanf("%d", &n);

  int a[n];
  printf("Enter the elements of the array: ");

  for (i = 0; i < n; i++) {
    scanf("%d", &a[i]);
  }
  max = a[0];
  min = a[0];
  for (i = 1; i < n; i++) {
    if (a[i] > max) {
      max = a[i];
    }
    if (a[i] < min) {
      min = a[i];
    }
  }
  printf("Maximum element in the array: %d\n", max);
  printf("Minimum element in the array: %d\n", min);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\maxMinArra...

```
Enter the number of elements in the array: 5
Enter the elements of the array: 2 7 1 9 5
Maximum element in the array: 9
Minimum element in the array: 1

Process returned 0 (0x0)   execution time : 8.209 s
Press any key to continue.
```

Recursion:

recursionFactorial.c | recursionFibonacci.c | stringLength.c | stringReverse.c | structure.c | switch.c | parameterArgument.c

```c
#include <stdio.h>

//factorial using recursion

int factorial(int n)
{
    if (n == 0)
        return 1;
    return n * factorial(n - 1);
}

int main()
{
    int num;
    printf("Enter a number to find its factorial: ");
    scanf("%d", &num);
    printf("Factorial is: %d", factorial(num));
    return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\recursionFactorial.e...

```
Enter a number to find its factorial: 6
Factorial is: 720
Process returned 0 (0x0)   execution time : 3.878 s
Press any key to continue.
```

recursionFibonacci.c | stringLength.c | stringReverse.c | structure.c | switch.c | parameterArgument.c

```c
#include <stdio.h>

//fibonacci series using recursion

int fibonacci(int n)
{
    if (n == 0)
        return 0;
    if (n == 1)
        return 1;
    return fibonacci(n - 1) + fibonacci(n - 2);
}

int main()
{
    int num, i;
    printf("Enter the amount of numbers: ");
    scanf("%d", &num);
    printf("Fibonacci series of %d numbers: \n", num);
    for (i = 0; i < num; i++)
    {
        printf("%d ", fibonacci(i));
    }
    return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\recursion...

```
Enter the amount of numbers: 9
Fibonacci series of 9 numbers:
0 1 1 2 3 5 8 13 21
Process returned 0 (0x0)   execution time : 6.020 s
Press any key to continue.
```

String:

stringLength.c × stringReverse.c × structure.c × switch.c × parameterArgument.c ×

```c
#include <stdio.h>
#include <string.h>

//string reverse with strrev

int main() {
  char str[100];
  printf("Enter a string: ");
  gets(str);
  strrev(str);
  printf("Reversed string: %s", str);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final...

```
Enter a string: haaland
Reversed string: dnalaah
Process returned 0 (0x0)   execution time : 13.872 s
Press any key to continue.
```

stringLength.c × structure.c × switch.c × parameterArgument.c ×

```c
#include <stdio.h>
#include <string.h>

//string length with strlen

int main() {
  char str[100];
  int length;
  printf("Enter a string: ");
  gets(str);
  length = strlen(str);
  printf("Length of the string is: %d", length);
  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignment...

```
Enter a string: alvarez
Length of the string is: 7
Process returned 0 (0x0)   execution time : 10.855 s
Press any key to continue.
```

Structure:

structure.c × switch.c × parameterArgument.c ×

```c
#include <stdio.h>
#include <string.h>

//structure example

struct student {
  int roll;
  int age;
  float cgpa;
};

int main() {
  struct student s1;
  s1.roll = 1061;
  s1.age = 20;
  s1.cgpa = 2.55;

  printf("Student Details: \n");
  printf("Roll: %d\n", s1.roll);
  printf("Age: %d\n", s1.age);
  printf("CGPA: %.2f\n", s1.cgpa);

  return 0;
}
```

"G:\C drive backup 221227\University Assignments\Cse assignments\Cse final\structur...

```
Student Details:
Roll: 1061
Age: 20
CGPA: 2.55

Process returned 0 (0x0)   execution time : 0.038 s
Press any key to continue.
```