2.1. Bài tập 1

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình in bảng cửu chương N do người dùng nhập vào.
   a) Xây dựng chương trình con:
      - Hàm **BangCuuChuong** truyền vào một tham số để xuất bảng cửu chương của số đó. Hàm không có giá trị trả về. `void BangCuuChuong(int n)`
   b) Tại chương trình chính:
      - Khai báo biến X, yêu cầu người dùng nhập vào giá trị muốn xuất bảng cửu chương.
      - Gọi hàm **BangCuuChuong** và truyền vào tham số thực X.

2.2. Bài tập 2

1. **Thời lượng:** 15 phút.
1. **Mô tả bài toán:** Viết chương trình nhập vào một số và tính bình phương, căn bậc hai của số đó theo yêu cầu sau:
   a) Xây dựng chương trình con:
      - Hàm **BinhPhuong** truyền vào một tham số để tính bình phương của số đó, rồi trả về kết quả bình phương của số đó. `int BinhPhuong (int so)`
      - Hàm **CanBacHai** truyền vào một tham số để tính căn bậc hai của số đó, rồi trả về kết quả căn bậc hai của số đó. `float CanBacHai (int so)`
   b) Tại chương trình chính:
      - Yêu cầu nhập số.
      - Gọi hàm **BinhPhuong** và truyền vào số vừa nhập để tính bình phương. Xuất kết quả ra màn hình.
      - Gọi hàm **CanBacHai** và truyền vào số vừa nhập để tính căn bậc hai. Xuất kết quả ra màn hình.

2.3. Bài tập 3

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập vào 2 số nguyên a, b. Tạo menu chức năng với mỗi chức năng xây dựng một hàm gồm:
   1. Tính tổng a + b
   2. Tính hiệu a – b
   3. Tính tích a * b
   4. Tính thương a / b
3. **Gợi ý:**
   - Các hàm có truyền 2 tham số A, B và có trả về kết quả của phép tính đó.
   - Tại chương trình chính, sử dụng lệnh switch để phân phối công việc.

2.4. Bài tập 4

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập vào số nguyên dương N. Kiểm tra N có là số nguyên tố hay không. Nếu có, thì xuất các số nguyên tố nhỏ hơn N.
   a) Xây dựng chương trình con:
      - Hàm **KiemTraSNT** có truyền vào tham số N, hàm thực hiện kiểm tra và trả về kết quả 1 nếu là số nguyên tố và 0 nếu số không là nguyên tố.
      - Hàm **LietKeSNT** có truyền vào tham số N thực hiện hiển thị các số nguyên tố nhỏ hơn N. Hàm không có giá trị trả về.

   b) Tại chương trình chính:
      - Yêu cầu nhập số.
      - Gọi hàm **KiemTraSNT** và so sánh kết quả:
        - Nếu là 0 thì xuất nội dung "*N không phải là số nguyên tố*".
        - Ngược lại, xuất nội dung "*N là số nguyên tố*", đồng thời gọi hàm **LietKeSNT.**
3. **Gợi ý:** Số nguyên tố là số chỉ chia hết cho 1 và cho chính nó.

2.5. Bài tập 5

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập số nguyên dương A, B, C. Thực hiện bài toán tính tổng của các giai thừa A, B và C: **S = A! + B! + C!**
   a) Xây dựng chương trình con:
      - Hàm **NhapSoNguyen** nhập vào số nguyên dương. Nếu là số âm thì yêu cầu nhập lại. Hàm không truyền tham số. Hàm trả về giá trị là 1 số nguyên dương. **int NhapSoNguyen()**
      - Hàm **GiaiThua** có truyền 1 tham số nguyên, hàm trả về giá trị giai thừa của số nguyên đó. **int GiaiThua(int n)**

   b) Tại chương trình chính:
      - Khai báo 3 biến: A, B, C, tổng giai thừa.
      - Giá trị của ba biến A, B, C là phép gán từ việc gọi hàm **NhapSoNguyen**.
      - Cộng dồn biến tổng giai thừa từ việc gọi hàm **GiaiThua** 3 lần có truyền tuần tự 3 biến A, B và C.
      - Xuất kết quả biến tổng giai thừa.
3. **Ví dụ:**
      - **Input**: A=1, B=3, C=5.
      - **Output**: Tổng giai thừa = 127.

2.6. Bài tập 6

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập vào số nguyên theo yêu cầu sau:
   a) Xây dựng chương trình con:
      - Hàm **TongSo** truyền vào tham số để tính theo cấu trúc 1+2+3+4+…+n, và tham chiếu kết quả KetQuaTongSo để lưu lại giá trị phép tính sau khi đã tính toán xong. Hàm không có giá trị trả về.
      - Hàm **TongHieuXenKe** truyền vào tham số để tính theo cấu trúc 1-2+3-4+…+n, và tham chiếu kết quả KetQuaTongHieuXenKe để lưu lại giá trị phép tính sau khi đã tính toán xong. Hàm không có giá trị trả về.

   b) Tại chương trình chính:
      - Khai báo biến KetQuaTongSo và biến KetQuaTongHieuXenKe.
      - Nhập vào số nguyên.
      - Gọi hàm **TongSo** có truyền tham số số nguyên vừa nhập và tham chiếu KetQuaTongSo.
      - Gọi hàm **TongHieuXenKe** có truyền tham số số nguyên vừa nhập và tham chiếu KetQuaTongHieuXenKe.
      - Hiển thị kết quả của 2 phép toán ra màn hình.

2.7. Bài tập 7

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình tính can chi dựa vào số năm cho trước.
   a) Xây dựng chương trình con:
      - Hàm **TinhCan** lấy số cuối cùng của năm sinh và xét và trả về kiểu chuỗi ký tự theo bảng sau

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

      - Hàm **TinhChi** lấy năm sinh – 1800 rồi chia dư cho 12 và trả về kiểu chuỗi ký tự theo bảng sau

0	1	2	3	4	5
Thân	Dậu	Tuất	Hợi	Tí	Sửu
6	**7**	**8**	**9**	**10**	**11**
Dần	Mẹo	Thìn	Tị	Ngọ	Mùi

   b) Tại chương trình chính:
      - Nhập năm sinh
      - Hiển thị đầy đủ tên can và chi của một năm âm lịch ra màn hình.
3. **Ví dụ:**
      - **Input**: 1993.
      - **Output**: Quý Dậu.

2.8. Bài tập 8

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Viết chương trình tính diện tích và chu vi theo yêu cầu sau:
   a) Xây dựng chương trình con:
   - Hàm **DienTichHinhTron** truyền vào tham số bán kính để tính diện tích của hình tròn, rồi trả về kết quả diện tích. `float DienTichHinhTron (float bankinh)`
   - Hàm **ChuViHinhTron** truyền vào tham số bán kính để tính chu vi của hình tròn, rồi trả về kết quả chu vi. `float ChuViHinhTron (float bankinh)`
   - Hàm **DienTichHinhVuong** truyền vào tham số cạnh để tính diện tích của hình vuông, rồi trả về kết quả diện tích. `float DienTichHinhVuong (int canh)`
   - Hàm **ChuViHinhVuong** truyền vào tham số cạnh để tính chu vi của hình vuông, rồi trả về kết quả chu vi. `float ChuViHinhVuong (int canh)`
   - Hàm **DienTichHCN** truyền tham số chiều dài, chiều rộng để tính diện tích hình chữ nhật, rồi trả về kết quả diện tích. `float DienTichHCN (int CD, int CR)`
   - Hàm **ChuViHCN** truyền vào tham số chiều dài và chiều rộng để tính chu vi của hình chữ nhật, rồi trả về kết quả chu vi. `float ChuViHCN (int CD, int CR)`
   - Hàm **XuatKetQua** truyền vào 3 tham số gồm hình dạng, kết quả chu vi, kết quả diện tích. Hàm **XuatKetQua** không có giá trị trả về. Hàm dựa vào tham số hình dạng (1: Hình tròn, 2: Hình vuông, 3: Hình chữ nhật) để xuất ra màn hình cấu trúc nội dung như sau:
     - Chu vi hinh XXXX la: YY
     - Dien tich hinh XXXX la: YY
     `void XuatKetQua (int hinhdang, float CV, float DT)`

   b) Tại chương trình chính:
   - Hiển thị lựa chọn:
     1. Hình tròn: Nhập bán kính.
     2. Hình vuông: Nhập cạnh.
     3. Hình chữ nhật: Nhập cạnh a và cạnh b.
   - Nếu chọn số 1:
     - Gọi hàm **ChuViHinhTron** truyền vào số vừa nhập để tính chu vi.
     - Gọi hàm **DienTichHinhTron** truyền vào số vừa nhập để tính diện tích.
   - Nếu chọn số 2:
     - Gọi hàm **ChuViHinhVuong** truyền vào số vừa nhập để tính chu vi.
     - Gọi hàm **DienTichHinhVuong** truyền số vừa nhập để tính diện tích.
   - Nếu chọn số 3:
     - Gọi hàm **ChuViHCN** truyền vào 2 số vừa nhập để tính chu vi.
     - Gọi hàm **DienTichHCN** truyền vào 2 số vừa nhập để tính diện tích.
   - Gọi hàm **XuatKetQua** để hiển thị kết quả.

2.9. Bài tập 9

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình xây dựng hàm tìm ước chung lớn nhất giữa 2 số nguyên A và B.
   a) Xây dựng chương trình con:
      - Hàm **NhapSoNguyen** yêu cầu người dùng nhập vào một số nguyên. Nếu là số âm thì hàm triệt tiêu dấu âm và trả về kết quả.
      - Hàm **TimUCLN** truyền vào 2 tham số a và b. Hàm trả về kết quả là ước số chung lớn nhất của 2 tham số a,b.

   b) Tại chương trình chính:
      - Khai báo hai biến a, b.
      - Lần lượt gán giá trị a và b bằng cách gọi hàm **NhapSoNguyen**.
      - Gọi hàm **TimUCLN** truyền 2 tham số và hiển thị kết quả.

2.10. Bài tập 10

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập hai số A và B. Hoán vị giá trị của hai số này.
   a) Xây dựng chương trình con:
      - Hàm **HoanVi** truyền vào 2 tham chiếu số nguyên A, B. Hàm thay đổi trực tiếp giá trị trên 2 tham số này. Hàm không có giá trị trả về.
        `void HoanVi(int &a, int &b)`
      - Hàm **XuatGiaTri** truyền vào 2 tham số ký tự X chứa tên của biến và số nguyên Y chứa giá trị của biến đó. Hàm xuất nội dung *"Biến X có giá trị Y"*. Hàm không có giá trị trả về. `void XuatGiaTri(char a, int n)`

   b) Tại chương trình chính:
      - Yêu cầu nhập vào 2 số a, b.
      - Gọi hàm **XuatGiaTri** 2 lần và truyền vào tuần tự 2 tham số thực a và b trước khi hoán vị.
      - Gọi hàm **HoanVi** có truyền vào 2 tham số thực a và b.
      - Gọi lại hàm **XuatGiaTri** để xem kết quả sau khi hoán vị.
3. **Gợi ý:** Thuật giải hoán vị
   - Tạo thêm một biến tạm. ➔ `int temp;`
   - Gán tạm bằng giá trị của 1 trong 2 biến a hoặc b. ➔ `temp = a;`
   - Biến vừa gán giá trị tiếp nhận giá trị mới của biến còn lại. ➔ `a = b;`
   - Biến còn lại nhận giá trị mới từ biến tạm. ➔ `b = temp;`

2.11. Bài tập 11

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình giải phương trình bậc hai $Ax^2 + Bx + C = 0$.
   a) Xây dựng chương trình con:
   - Hàm **GiaiPhuongTrinh** có truyền 5 tham số gồm 3 tham trị A, B, C và 2 tham chiếu X1 và X2, với A, B, C là hệ số phương trình và X1, X2 là nghiệm của phương trình. Hàm trả về kết quả là một giá trị số nguyên có ý nghĩa như sau:

Giá trị	Ý nghĩa
-1	Phương trình vô nghiệm.
0	Phương trình vô số nghiệm.
1	Phương trình có 1 nghiệm kép. Đồng thời, lưu kết quả vào 2 tham chiếu.
2	Phương trình có 2 nghiệm phân biệt. Đồng thời, lưu kết quả vào 2 tham chiếu.

   b) Tại chương trình chính:
   - Yêu cầu nhập 3 hệ số a, b, c.
   - Khai báo 2 biến chứa nghiệm phương trình.
   - Gọi hàm **GiaiPhuongTrinh** và truyền vào 5 tham số thực.
   - Xuất kết quả bài toán.

2.12. Bài tập 12

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình chính nhập số nguyên lớn hơn 2 chữ số và làm theo yêu cầu sau.
   a) Xây dựng chương trình con:
   - Hàm **KiemTraChuSoToanChan** truyền vào 1 tham số để kiểm tra từng ký tự số có là số chẵn. Nếu tất cả ký tự số cùng chẵn thì hàm trả về giá trị 1. Ngược lại, chỉ cần bất kỳ ký tự số là số lẻ thì hàm trả về giá trị 0.
   - Hàm **KiemTraTangDan** truyền vào 1 tham số để kiểm tra từng ký tự số từ trái qua phải có tăng dần hay không. Nếu có hàm trả về giá trị 1. Ngược lại, thì hàm trả về giá trị 0.
   - Hàm **DemSoChanLe** truyền vào 3 tham số gồm: 1 tham trị số cần kiểm tra và 2 tham chiếu đếm số chẵn và đếm số lẻ. Hàm duyệt từ trái sang phải và đếm tất cả ký tự số chẵn, và đếm tất cả ký tự số lẻ. Hàm không có giá trị trả về. VD: n=5263 có 2 chữ số lẻ (5 và 3), 2 chữ số chẵn (2 và 6).
     `void DemSoChanLe(int songuyen, int &DemChan, int &DemLe)`
   b) Tại chương trình chính:
   - Khai báo biến songuyen, DemChan, DemLe.
   - Nhập vào số nguyên.

- Gọi hàm **KiemTraChuSoToanChan** có truyền tham số songuyen vừa nhập
  - Nếu hàm trả về 1: Xuất thông báo "Số của bạn là số chứa toàn các ký tự chẵn".
  - Nếu hàm trả về 0: Xuất thông báo "Số của bạn có chứa ký tự lẽ".
- Gọi hàm **KiemTraTangDan** có truyền tham số songuyen vừa nhập
  - Nếu hàm trả về 1: Xuất thông báo "Số %d là số có các ký tự số tăng dần".
  - Nếu hàm trả về 0: Xuất thông báo "Số %d là số không có các ký tự số tăng dần".

Câu hỏi ôn tập

(Xem đáp án ở trang)

3.1. Trong các phát biểu sau về hàm, phát biểu nào là đúng?

A. Một hàm nhất thiết phải có danh sách tham số.
B. Một hàm nhất thiết phải có biến cục bộ.
C. Một hàm nhất thiết phải có từ khoá return.
D. Một hàm không nhất thiết phải có danh sách tham số, biến cục bộ hay return.

3.2. Để định dạng tham chiếu, bạn thêm ký hiệu nào trước tên tham số?

A. $
B. #
C. &
D. @

3.3. What are mandatory parts in the function declaration?

A. return type, function name.
B. parameters, function name.
C. return type, function name, parameters.
D. parameters, variables.

3.4. Hãy cho biết kết quả đoạn code sau:

```
void ChangeValue(int a, int b)
{
    a=6;
    b=a+1;
}
void main()
{
    int a=3, b=5;
    ChangeValue(a,b);
    printf("A= %d, B= %d", a, b);
}
```

A. A= 3, B= 5
B. A= 6, B= 6
C. A= 6, B= 7
D. A= 3, B= 4

3.5. Hãy cho biết kết quả của đoạn code sau:

```
void swap(int &a, int &b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
    printf("In swap: %d – %d. ", a, b);
}
void main()
{
    int a = 5, b = 10;
    swap(a, b);
    printf("In main: %d – %d. ", a, b);
}
```

A. In swap: 10 – 5. In main: 10 – 5.
B. In swap: 10 – 5. In main: 5 – 10.
C. In swap: 5 – 10. In main: 10 – 5.
D. In swap: 5 – 10. In main: 5 – 10.

3.6. Hãy cho biết hàm print ở main được gọi từ hàm nào?

```
void print(int i)
{
    printf("%d", i);
}
void print(float f)
{
    printf("%f", f);
}
void main()
{
    print(5.75F);
}
```

A. void print(int i) ;
B. void print(float f);
C. Compile error vì hàm trùng tên.

3.7. Hãy cho biết kết quả của đoạn code sau?

```
int Divide(int X, int Y, int Z)
{
    return X / Y;
}
float Divide(double X, double Y, double Z)
{
    return X / Y;
}
void main()
{
    float Result = Divide(5, 2);
    printf("%.2f", Result);
}
```

A. 2.00
B. 2.50
C. Compile error vì hàm trùng tên.
D. Compile error vì hàm không tìm thấy danh sách tham số.

3.8. Hãy cho biết kết quả của đoạn code sau:

```
void fun(int x, int &y)
{
    x = 20;
    y = 10;
}
void main()
{
    int x = 5;
    fun(x, x);
    printf("%d", x);
}
```

A. 20 B. 10 C. 5 D. Compile error

3.9. Hãy cho biết kết quả của đoạn code sau:

```
int max(int a, int b)
{
    return ( a > b ? a : b );
}
void main()
{
    int i = 5;
    int j = 7;
    printf("%d", max(i, j));
}
```

A. 5 B. 7 C. 0 D. Một giá trị khác.

3.10. Hãy cho biết kết quả của đoạn code sau:

```
int mult(int x, int y)
{
    int result;
    result = 0;
    while (y != 0)
    {
        result = result + x;
        y = y - 1;
    }
    return result;
}
void main()
{
    int x = 5, y = 5;
    int result = mult(x, y);
    printf("%d", result);
}
```

A. 20 C. 30

B. 25 D. 35

3.11. Hãy cho biết kết quả của đoạn code sau:

```
int add(int first, int second)
{
    return first + second + 15;
}
int operation(int a, int b, int (*functocall)(int, int))
{
    return (*functocall)(a, b);
}
int main()
{
    int result;
    int (*plus)(int, int) = add;
    result = operation(15, 10, plus);
    printf("%d", result);
}
```

A. 25 B. 35 C. 40 D. 45

3.12. Hãy cho biết kết quả của đoạn code sau:

```
int fun(int=0, int = 0);
void main()
{
  printf("%d", fun(5));
}
int fun(int x, int y)
{
    return (x+y);
}
```

A. -5 B. 0 C. 10 D. 5