2.1. Bài tập 1

1. **Thời lượng:** 20 phút.
2. **Mô tả bài toán:** Khai báo một mảng lưu trữ các số nguyên có giá trị {10, 25, -4, 32, 63, 81, 19, -24, 13, 18, 45, 12, 72, 42, -6}. Viết chương trình thực hiện các yêu cầu sau:
   - **Yêu cầu 1:** Liệt kê vị trí và giá trị các phần tử là số chẵn.
   - **Yêu cầu 2:** Đếm các phần tử có giá trị chia hết cho 3 và cũng chia hết cho 9.
   - **Yêu cầu 3:** Tính tổng giá trị các phần tử có trị số chẵn.
   - **Yêu cầu 4:** Tìm kiếm và hiển thị trị số của các phần tử có giá trị âm. Thay thế các phần tử có giá trị âm bằng giá trị 0.

2.2. Bài tập 2

1. **Thời lượng:** 20 phút.
2. **Mô tả bài toán:** Khai báo một mảng lưu trữ số cân nặng (kg) của 20 ứng viên tham gia nghĩa vụ quân sự gồm các giá trị {36.5, 98, 27.8, 63, 78.1, 48.3, 69, 72, 41.5, 32, 29.5, 120, 52.3, 23, 50.2, 56, 72.5, 70, 68.4, 65}. Chỉ số cân nặng của một người bình thường dao động từ 38 kg đến 75kg. Viết chương trình thực hiện các yêu cầu sau:
   - **Yêu cầu 1:** Liệt kê danh sách các ứng viên thiếu cân, thừa cân.
   - **Yêu cầu 2:** Tìm kiếm ứng viên có cân nặng thấp nhất, cao nhất.
   - **Yêu cầu 3:** Đếm số lượng ứng viên đạt tiêu chuẩn.
   - **Yêu cầu 4:** Tính số cân nặng trung bình của các ứng viên đạt tiêu chuẩn.

2.3. Bài tập 3

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Khai báo một mảng các số nguyên. Viết chương trình thực hiện các yêu cầu sau:
   - **Yêu cầu 1:** Yêu cầu người dùng nhập vào giá trị A. Tìm kiếm các phần tử ở trị số chẵn có giá trị nhỏ hơn hoặc bằng A.
   - **Yêu cầu 2:** Yêu cầu người dùng nhập vào giá trị B. Đếm số lần xuất hiện của các phần tử mang giá trị trùng với giá trị B.
   - **Yêu cầu 3:** Yêu cầu người dùng nhập vào 2 giá trị X, Y. Liệt kê các phần tử có giá trị nằm trong phạm vi từ X đến Y.

2.4. Bài tập 4

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Khai báo một mảng các số nguyên. Viết chương trình yêu cầu người dùng nhập vào giá trị các phần tử. Kiểm tra mảng chứa các phần tử có giá trị đối xứng nhau hay không
3. **Ví dụ:**
   - **Input:** [3 2 1 2 3] → **Output**: Đây là một mảng đối xứng.
   - **Input**: [1 2 4 4 5 1] → **Output**: Đây là một mảng không đối xứng.

2.5. Bài tập 5

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Viết chương trình khai báo mảng số nguyên có tối đa 20 phần tử.
   a) Xây dựng chương trình con:
      - Hàm **NhapMang** truyền vào tham số số lượng phần tử muốn sử dụng và mảng. Hàm yêu cầu người dùng nhập giá trị phần tử và lưu vào mảng. Hàm không có giá trị trả về. `void NhapMang(int &Soluong, int Arr[])`
      - Hàm **XuatMang** truyền vào tham số số lượng phần tử đang sử dụng và mảng. Hàm xuất giá trị của tất cả phần tử có trong mảng. Hàm không có giá trị trả về. `void XuatMang(int Soluong, int Arr[])`
      - Hàm **HoanVi** truyền vào 2 tham số số nguyên. Hàm thực hiện đảo giá trị của 2 tham số. Hàm không có giá trị. `void HoanVi(int &a, int &b)`
      - Hàm **BubbleSort** truyền vào tham số mảng. Hàm áp dụng thuật giải Bubble Sort để sắp xếp giá trị các phần tử theo trật tự tăng dần. Hàm này có sử dụng hàm **HoanVi** khi cần thiết. Hàm không có giá trị trả về.
        `void BubbleSort (int Soluong, int Arr[])`
      - Hàm **SelectionSort** truyền vào tham số mảng. Hàm áp dụng thuật giải Selection Sort để sắp xếp giá trị các phần tử theo trật tự giảm dần. Hàm này có sử dụng hàm **HoanVi** khi cần thiết. Hàm không có giá trị trả về.
        `void SelectionSort (int Soluong, int Arr[])`

   b) Tại chương trình chính:
      - Khai báo mảng, biến số lượng phần tử sử dụng.
      - Yêu cầu người dùng nhập số lượng phần tử muốn sử dụng. Cần đảm bảo số lượng người dùng nhập vào phải ít hơn hoặc bằng số lượng phần tử tối đa được khai báo trong mảng.
      - Gọi hàm **BubbleSort** truyền vào mảng đã khai báo để sắp xếp trật tự tăng dần.
      - Gọi hàm **XuatMang** truyền vào mảng đã khai báo và số lượng phần tử sử dụng để xem kết quả sau khi sắp xếp.
      - Gọi hàm **SelectionSort** truyền vào mảng đã khai báo để sắp xếp trật tự giảm dần.
      - Gọi hàm **XuatMang** truyền vào mảng đã khai báo và số lượng phần tử sử dụng để xem kết quả sau khi sắp xếp.

2.6. Bài tập 6

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Khai báo mảng có tối đa 6 phần tử gồm các giá trị {2, 3, 4, 5}. Viết chương trình thực hiện các yêu cầu sau:
   - **Yêu cầu 1:** Thêm giá trị 6 vào phần tử ở vị trí cuối mảng.
   - **Yêu cầu 2:** Thêm giá trị 1 vào phần tử ở vị trí đầu mảng.

2.7. Bài tập 7

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Khai báo mảng toàn cục có tối đa 10 phần tử gồm các giá trị {1, 2, 3, 4, 5, 6, 7, 8, 9}. Biến số lượng phần tử đang sử dụng là 9 cũng được khai báo toàn cục.
   a) Xây dựng chương trình con:
      - Hàm **AddValue** truyền vào 2 tham số gồm giá trị và vị trí phần tử cần thêm vào mảng. Sau khi thêm, tăng biến số lượng phần tử 1 đơn vị. Hàm không có giá trị trả về. `void AddValue(int value, int pos)`
      - Hàm **RemoveValue** truyền vào tham số vị trí phần tử cần xoá ra khỏi mảng. Sau khi xoá, giảm biến số lượng phần tử 1 đơn vị Hàm không có giá trị trả về. `void RemoveValue(int pos)`
      - Hàm **OutputArray** hiển thị giá trị từng phần tử có trong mảng. Hàm không có giá trị trả về. `void OutputArray()`

   b) Tại chương trình chính:
      - Khai báo biến giá trị và vị trí.
      - Yêu cầu người dùng nhập vào giá trị và vị trí cần thêm.
      - Gọi hàm **AddValue** và truyền vào 2 biến giá trị và vị trí.
      - Gọi hàm **OutputArray** để hiển thị kết quả sau khi thêm.
      - Yêu cầu người dùng nhập vào vị trí phần tử cần xoá. (Quy định với người dùng vị trí đầu tiên trong mảng là 0).
      - Gọi hàm **RemoveValue** và truyền vào biến vị trí.
      - Gọi hàm **OutputArray** để hiển thị kết quả sau khi xoá.

2.8. Bài tập 8

1. **Thời lượng:** 30 phút.
2. **Mô tả bài toán:** Viết chương trình nhập/xuất mảng a các số nguyên có n phần tử bàn phím. Điều kiện giá trị của n phải thỏa mãn $1 \leq n \leq 100$. Nếu không thỏa mãn điều kiện trên thì yêu cầu nhập lại. Xây dựng chương trình con:
   - Viết hàm trả về giá trị và chỉ số của phần tử lớn nhất trong mảng.
   - Viết hàm trả về giá trị và chỉ số của phần tử nhỏ nhất trong mảng.
   - Viết hàm trả về vị trí đầu tiên của số x xuất hiện trong mảng.
   - Viết hàm trả về vị trí cuối cùng của số x xuất hiện trong mảng.
   - Viết hàm đếm số lần xuất hiện của phần tử x trong mảng.

2.9. Bài tập 9

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Cho mảng a gồm n số nguyên. Viết chương trình sắp xếp giá trị các phần tử có trị số chẵn trong mảng theo thứ tự tăng dần.
3. **Gợi ý:** Thay đổi bước nhảy của vòng lặp khi thực hiện sắp xếp trong lúc duyệt mảng. `for(int i=0; i<n; i+=2)`

2.10. Bài tập 10

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Viết chương trình nhập/xuất mảng a các số nguyên có n phần tử nhập từ bàn phím. Điều kiện giá trị của n phải thỏa mãn $1 \leq n \leq 100$. Nếu không thỏa mãn điều kiện trên thì yêu cầu nhập lại. Xây dựng chương trình con:
   - Hàm tính giá trị trung bình các phần tử có giá trị âm trong mảng. Hàm trả về giá trị trung bình của các phần tử có giá trị âm. Ngược lại hàm trả về giá trị 0.
   - Hàm tính giá trị trung bình các phần tử có giá trị dương trong mảng. Hàm trả về giá trị trung bình của các phần tử có giá trị dương. Ngược lại hàm trả về giá trị 0.
   - Hàm tính giá trị trung bình các số chẵn và chia hết cho 3 trong mảng. Hàm trả về giá trị trung bình của các phần tử thỏa mãn. Ngược lại hàm trả về giá trị 0.

2.11. Bài tập 11

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Cho mảng a gồm n số nguyên có thứ tự tăng dần. Viết chương trình chèn một số nguyên do người dùng nhập vào sao cho mảng vẫn có thứ tự tăng dần.
3. **Gợi ý:** Duyệt từ cuối mảng theo trình tự ngược, mỗi lần duyệt so sánh giá trị phần tử trong mảng với số nguyên nhập vào. Nếu phát hiện giá trị nào nhỏ hơn thì dừng lại. Di dời các phần tử từ vị trí phát hiện về sau 1 đơn vị. Chèn số nguyên và tăng số lượng phần tử lên 1.

2.12. Bài tập 12

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Cho mảng a gồm n số nguyên có giá trị toàn cục. Viết chương trình xoá tất cả các số chia hết cho 4 trong mảng.
   a) Xây dựng chương trình con:
      - Hàm **XoaPhanTu** có truyền tham số vị trí muốn xoá. Hàm thực hiện xoá phần tử tại vị trí cho trước. Hàm không có giá trị trả về.
      - Hàm **KiemTra** có truyền tham số là giá trị từng phần tử trong mảng. Hàm trả về giá trị 1 khi giá trị chia hết cho 4 và 0 cho trường hợp ngược lại.
      - Hàm **HienThi** hiển thị giá trị từng phần tử trong mảng.
   b) Tại chương trình chính:
      - Yêu cầu người dùng nhập vào giá trị số nguyên.
      - Duyệt từ đầu mảng, mỗi lần duyệt, gọi hàm **KiemTra** và truyền vào giá trị phần tử trong mảng. Nếu hàm **KiemTra** trả về giá trị 1 thì gọi hàm **XoaPhanTu** và truyền vào trị số cần xoá.
      - Sau quá trình duyệt mảng, gọi hàm **HienThi** để quan sát kết quả.

2.13. Bài tập 13

1. **Thời lượng:** 15 phút.
2. **Mô tả bài toán:** Cho mảng a gồm n số nguyên. Viết hàm tìm và xóa tất cả các phần tử trùng với x trong mảng một chiều a có n phần tử, nếu không tồn tại phần tử x trong mảng thì trả về -1.

Câu hỏi ôn tập

(Xem đáp án ở trang)

3.1. Để tạo một mảng các số nguyên, cách khai báo nào sau đây là đúng?

A. int []SoNguyen;
B. int SoNguyen[];
C. int SoNguyen[5];
D. int SoNguyen[]={1;2;3;4;5;6};

3.2. Khai báo mảng sau: int n[5] = { 32, 27, 14, 18, 95, 14 };

A. Mảng này là chính xác.
B. Compile Error: too many initializer values.
C. Runtime Error.
D. Lỗi xảy ra khi mảng có 2 giá trị 14 trùng nhau.

3.3. Khai báo mảng int x[3] = {4, 2, 6};. Hãy cho biết giá trị của từng phần tử trong mảng sau?

A. x[1] = 4; x[2] = 2; x[3] = 6;
B. x[0] = 4; x[1] = 2; x[2] = 6;
C. x[3] = 4; x[3] = 2; x[3] = 6;
D. x[4] = 3; x[2] = 3; x[6] = 3;

3.4. Cho mảng A[5]={7,4,1,3,8}. Muốn in giá trị cuối cùng trong mảng ra màn hình, bạn sử dụng lệnh nào?

A. printf("%d", A[5]);
B. printf("%d", A[4]);
C. printf("%d", A(5));
D. printf("%d", A(4));

3.5. Cho mảng A gồm 10 phần tử số nguyên. Bạn sử dụng lệnh nào để gán giá trị 6 cho phần tử đầu tiên trong mảng?

A. A(0) = 6;
B. A(1) = 6;
C. A[0] = 6;
D. A[1] = 6;

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

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

A. 1 và 4 giá trị rác.
B. 1 0 0 0 0
C. 1 1 1 1 1
D. 0 0 0 0 0

3.7. Cho các khai báo sau. Lệnh nào trong các phương án dưới đây không sai về cú pháp?

```
char msg[10];
char value;
```

A. msg[2] = value;
B. msg = value;
C. Cả hai câu trên.
D. Không câu nào đúng.

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

```
void main()
{
	int arr[]= {1,2,3,4,5};
	printf("&d and %d", arr[2], arr[4]);
}
```

A. 2 and 4
B. 4 and 2
C. 3 and 5
D. 5 and 3

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

```
void main()
{
	int egArray[] = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
	for (int index = 0; index < 5; index++)
	    printf("&d ", egArray[index]);
}
```

A. 2 4 6 8
B. 2 4 6 8 10
C. 2 4 6 8 10 1
D. 2 4 6 8 10 1 3 5 7 9

3.10. Để tất cả các giá trị bên trong mảng đều tăng gấp đôi. Bạn sẽ viết code như thế nào trong khoảng trống?

```
int array[10] = {2,4,8,3,4,5,1,9,7,0}
for (int index=0; index < 10; index++)
{	---???---	}
```

A. index = 2 * index;
B. array[2*index] = 2 * index;
C. array[index] = 2 * array[index];
D. array[index] = 2 * index;

3.11. Để các phần tử trong mảng được in ra theo trình tự *ĐẢO NGƯỢC*. Hãy cho biết các thành phần trong vòng lặp for được viết như thế nào?

```
void main()
{
	int egArray[] = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
	for ( --- ???--- )
	    printf("&d ", egArray[index]);
}
```

A. int index = 0 ; i < 10 ; i++
B. int index = 10 ; i < 0 ; i--
C. int index = 9 ; i > 0 ; i--
D. int index = 9 ; i >= 0 ; i--

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

```
void main()
{
       int egArray[] = { 2, 4, 6, 8, 10, 1, 3, 5, 7, 9 };
       for (int index = 0; index < 10; index+=2)
           printf("%d ", egArray[index]);
}
```

A. 2 4 6 8 10 1 3 5 7 9
B. 4 8 1 5 9
C. 2 6 10 3 7
D. 2 6 10 3 7 0

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

```
int array[] = { 1, 4, 3, 6, 8, 2, 5};
int what = array[0];
for (int index = 0; index < 7; index++)
{
   if (array[index] > what)
      what = array[index];
}
printf(“%d”, what);
```

A. 1
B. 5
C. 8
D. 1 4 3 6 8 2 5