SỞ GIÁO DỤC VÀ ĐẠO TÀO
THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG THPT NGUYỄN TẤT THÀNH

PHIẾU HƯỚNG DẪN HỌC SINH TỰ HỌC

(Đối với học sinh không thể học tập trực tuyến)

TIN HỌC 11 – TUẦN 06 (11/10/2021 đến 16/10/2021)

<table>
<tr><th>NỘI DUNG</th><th>GHI CHÚ</th></tr>
<tr><td>Tên bài học/ chủ đề</td><td>CHỦ ĐỀ 2: CHƯƠNG TRÌNH ĐƠN GIẢN
BÀI 6: PHÉP TOÁN, BIỂU THỨC, CÂU LỆNH GÁN</td></tr>
<tr><td>Hoạt động 1:
Đọc tài liệu và thực hiện các yêu cầu.
Học sinh đọc nội dung Bài 6 – SGK – Tin học 11, trang 24, 25, 26, 27, 28 để:
+Biết các khái niệm: phép toán, biểu thức số học, hàm số học chuẩn, biểu thức quan hệ, biểu thức lôgic;
+Hiểu và viết được lệnh gán trong Pascal;
+Viết được các biểu thức số học và lôgic với các phép toán thông dụng.
<u>Chú ý</u>:

1. Phép toán: HS cần chú ý một số phép toán mới như div, mod, not, and, or

2. Biểu thức số học: HS cần hiểu và nắm rõ các qui tắc viết biểu thức và thứ tự ưu tiên của các phép toán.</td><td>1. Phép toán
<table>
<tr><th>Phép toán</th><th>Trong toán học</th><th>Trong Pascal</th></tr>
<tr><td>Các phép toán số học với <u>số nguyên</u></td><td>+ (cộng), – (trừ), × (nhân), div (chia nguyên), mod (lấy phần dư)</td><td>+, –, *, div, mod</td></tr>
<tr><td>Các phép toán số học với <u>số thực</u></td><td>+ (cộng), – (trừ), × (nhân), : (chia)</td><td>+, –, *, /</td></tr>
<tr><td>Các <u>phép toán quan hệ</u></td><td>< (nhỏ hơn), ≤ (nhỏ hơn hoặc bằng), > (lớn hơn), ≥ (lớn hơn hoặc bằng), = (bằng), ≠ (khác)</td><td><, <=, >, >=, =, <></td></tr>
<tr><td>Các <u>phép toán lôgic</u></td><td>¬ (phủ định), ∧ (và), ∨ (hoặc)</td><td>not, and, or</td></tr>
</table>
<u>Chú ý</u>:
– Kết quả của các phép toán quan hệ cho giá trị <u>lôgic</u>
– Một trong những ứng dụng của phép toán lôgic là để tạo ra các biểu thức phức tạp từ các quan hệ đơn giản.
2. Biểu thức số học
▪ Trong Pascal, biểu thức số học là một biến kiểu số hoặc một hằng số hoặc các biến kiểu số và các hằng số liên kết với nhau bởi một số hữu hạn phép toán số học, các dấu ngoặc (và) tạo thành một biểu thức có dạng tương tự như cách viết trong toán học với những qui tắc sau:
• Chỉ dùng <u>cặp ngoặc tròn</u> để xác định trình tự thực hiện phép toán trong trường hợp cần thiết;
• Viết lần lượt từ trái sang phải;
• Không được bỏ qua dấu * trong tích.
▪ Các phép toán được thực hiện theo thứ tự:
• Thực hiện các phép toán <u>trong ngoặc</u> trước;
• Trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải, theo thứ tự các phép toán nhân (*), chia (/), chia lấy phần nguyên (div), lấy phần dư (mod) <u>thực hiện trước</u> và các phép toán cộng (+), trừ (–), <u>thực hiện sau</u>.
▪ <u>Chú ý</u>:
• Nếu biểu thức chứa một hằng hay biến kiểu thực thì ta có biểu thức số học thực, giá trị của biểu thức cũng thuộc kiểu thực.
• Trong một số trường hợp nên dùng biến trung gian để có thể tránh được việc tính biểu thức nhiều lần</td></tr>
</table>

NỘI DUNG	GHI CHÚ
3. **Hàm số học chuẩn:** HS biết ý nghĩa một số hàm số học chuẩn thường dùng trong Pascal để thực hiện chuyển đổi từ biểu thức toán học sang biểu thức trong Pascal.	**3. Hàm số học chuẩn** ▪ Hàm số học chuẩn **là những hàm *tính giá trị*** những hàm toán học thường dùng trong các ngôn ngữ lập trình. ▪ Mỗi hàm chuẩn có tên chuẩn riêng. **Đối số** của hàm là **một** hay **nhiều** biểu thức số học và được đặt trong **cặp ngoặc tròn** (và) sau tên hàm. ▪ Kết quả của hàm có thể là **nguyên** hoặc **thực** hay **phụ thuộc vào kiểu của đối số**. ▪ **Một số hàm chuẩn thường dùng:**

Hàm	Biểu diễn toán học	Biểu diễn trong Pascal	Kiểu đối số	Kiểu kết quả
Giá trị tuyệt đối	$\|x\|$	**abs(x)**	thực hoặc nguyên	theo kiểu của đối số
Bình phương	x^2	**sqr(x)**	thực hoặc nguyên	theo kiểu của đối số
Căn bậc hai	$\sqrt{x}$	**sqrt(x)**	thực hoặc nguyên	thực
Sin	**sinx**	**sin(x)**	thực	thực
Cos	**cosx**	**cos(x)**	thực	thực

***Ví dụ*:**

Biểu thức toán học	Biểu thức trong Pascal
$\frac{-b+\sqrt{b^2-4ac}}{2a}$	(-b+sqrt(b*b – 4*a*c))/(2*a) hoặc (-b+sqrt(sqr(b) – 4*a*c))/2/a
$\frac{x+y}{x-\frac{1}{2}} - \frac{x-z}{xy}$	(x+y)/(x-1/2)-(x-z)/(x*y)
$\frac{xy}{z}$	x*y/z

4. **Biểu thức quan hệ**: HS cần nắm rõ trình tự các bước thực hiện trong biểu thức quan hệ:
+ Tính giá trị các biểu thức;
+ Thực hiện phép toán quan hệ;
+ Cho kết quả biểu thức.

4. Biểu thức quan hệ
▪ Hai biểu thức **cùng kiểu** liên kết với nhau bởi **phép toán quan hệ** cho ta **một biểu thức quan hệ**.
Biểu thức quan hệ có dạng:
<biểu thức 1> <phép toán quan hệ> <biểu thức 2>
trong đó, *biểu thức 1* và *biểu thức 2* cùng là xâu hoặc cùng là biểu thức số học
▪ **Biểu thức quan hệ được thực hiện theo trình tự**:
- Tính giá trị các biểu thức;
- Thực hiện phép toán quan hệ.

▪ **Kết quả** của **biểu thức quan hệ** là giá trị **lôgic**: ***true*** (**đúng**) hoặc ***false*** (**sai**).
Ví dụ:
```
x < 5
i+1 >= 2*j
```

5. **Biểu thức lôgic**: HS cần viết đúng các biểu thức đơn giản trong chương trình, không nhầm lẫn với cách viết biểu thức trong toán học.

5. Biểu thức lôgic
▪ Biểu thức **lôgic** đơn giản là biến **lôgic** hoặc hằng **lôgic**.
▪ Biểu thức **lôgic** là các biểu thức **lôgic** đơn giản, các biểu thức quan hệ liên kết với nhau bởi phép toán **lôgic**. Giá trị biểu thức **lôgic** là ***true*** ***(1)*** hoặc ***false*** ***(0)***. Các biểu thức quan hệ thường được đặt trong **cặp ngoặc** **(** và **)**

NỘI DUNG	GHI CHÚ
	▪ Phép toán ***not*** được viết trước biểu thức cần phủ định ***Ví dụ***: `not (x<1) ⇔ x >= 1` ▪ Các phép toán ***and*** và ***or*** dùng để kết hợp nhiều biểu thức **lôgic** hoặc quan hệ thành một biểu thức, thường được dùng để diễn tả các *điều kiện* phức tạp. ▪ Bảng giá trị phép toán **lôgic**: **A** | **not A**: 1 | 0; 0 | 1 **A** | **B** | **A and B** | **A or B**: 1 1 1 1; 1 0 0 1; 0 1 0 1; 0 0 0 0 **Cần nhớ**: • A **and** B chỉ **đúng** khi cả 2 (A, B) cùng **đúng** • A **or** B chỉ **sai** khi cả 2 (A, B) cùng **sai** • A **or** B **đúng** khi có **ít nhất một** (mệnh đề) đúng ***Ví dụ* 1:** Để thể hiện ***điều kiện*** $5 \le x \le 11$, trong Pascal cần phải tách thành phát biểu dưới dạng "5≤x và x ≤11": $(5 <= x)$ and $(x <= 11)$ ***Ví dụ* 2:** Giả thiết **M** và **N** là hai biến nguyên. *Điều kiện* xác định **M** và **N** ***đồng thời*** chia hết cho 3 **hay** ***đồng thời*** không chia hết cho 3 được thể hiện trong Pascal như sau: `((M mod 3 = 0) and (N mod 3 = 0)) or ((M mod 3 <> 0) and (N mod 3 <>0))`
6. **Câu lệnh gán**: HS cần học và hiểu rõ cú pháp câu lệnh gán: <tên biến> := <biểu thức>; Câu lệnh có chức năng gán giá trị cho một biến, nghĩa là thay giá trị cũ trong ô nhớ (tương ứng với biến) bởi giá trị mới. Giá trị mới là giá trị của một biểu thức. Biểu thức này đã có giá trị xác định thuộc phạm vi của biến. HS cần lưu ý "**một số điểm cần chú ý**" khi sử dụng câu lệnh gán trong bài học.	**6. Câu lệnh gán** ▪ **Lệnh gán** là một trong những lệnh cơ bản nhất của các ngôn ngữ lập trình. ▪ Lệnh gán trong Pascal có dạng: ***<tên biến> := <biểu thức>;*** ▪ Trong trường hợp đơn giản, tên biến là tên của biến đơn. ▪ **Kiểu** của **giá trị biểu thức** phải phù hợp với **kiểu của biến**. ▪ **Chức năng của lệnh gán** là đặt cho biến có tên **ở vế trái** dấu **":="** giá trị mới bằng với giá trị của của biểu thức **ở vế phải**. ▪ Một biến **chỉ được coi là đã xác định giá trị** khi đã nhận được giá trị từ ngoài (đọc từ bàn phím hoặc từ tệp …) hoặc trực tiếp qua lệnh gán trong chương trình. ***Ví dụ:*** `i := i + 1;` {tăng i lên 1 đơn vị} `s := s + i;` ▪ **Một số điểm cần chú ý khi sử dụng lệnh gán:** • Phải viết **đúng** kí hiệu lệnh gán. ***Ví dụ***, trong Pascal kí tự **dấu hai chấm** phải **viết liền** kí tự **dấu bằng** (**:=**) • Biểu thức bên vế phải cần được xác định giá trị trước khi gán, nghĩa là mọi biến trong biểu thức đã được xác định giá trị và các phép toán trong biểu thức có thể thực hiện được trong miền giá trị của biến. • **Kiểu của biến** phải phù hợp với ***kiểu dữ liệu* của giá trị biểu thức bên vế phải**.

NỘI DUNG	GHI CHÚ
Hoạt động 2: ***Kiểm tra, đánh giá quá trình tự học.*** Học sinh đọc nội dung Bài 6 – SGK – Tin học 11, từ trang 24 đến trang 28 và trả lời các câu hỏi trắc nghiệm:	**Câu 1**: Trong **Pascal**, các biểu diễn của phép toán số học với **số nguyên** là: A. +, -, × , : B. +, -, *, div (chia lấy nguyên), mod (chia lấy dư) C. +, -, *, / D. +, -, × , div (chia lấy nguyên), mod (chia lấy dư) **Câu 2**: x là biến kiểu **integer**, y là biến kiểu **char**. Nếu thực hiện phép toán **x div y** ta sẽ có kết quả: A. x div y = 0 B. x div y < > 0 C. không thực hiện được phép tính vì x, y khác kiểu dữ liệu D. x div y >= 0 **Câu 3**: Chọn phát biểu **SAI** trong các phát biểu sau đây: A. Hầu hết các ngôn ngữ lập trình đều có các phép toán số học và phép toán quan hệ; B. Trong Pascal, phép chia số thực (kí hiệu là "/") cũng áp dụng được cho chia hai số nguyên; C. Trong máy tính, không thể chia một số cho số nhỏ tùy ý (tùy ý sát gần giá trị 0); D. Trong Pascal, phép chia số nguyên (kí hiệu là div) cũng áp dụng được cho hai số thực; **Câu 4**: Phát biểu nào dưới đây là **SAI**? A. Phép toán được thực hiện theo thứ tự từ trái qua phải; B. Để tính giá trị biểu thức, các biến và hằng trong biểu thức phải được xác định giá trị trước; C. Trong biểu thức số học, cặp ngoặc tròn "(" ")" khi cần thiết được dùng để xác định trình tự thực hiện phép toán; D. Thực hiện phép toán trong ngoặc trước. Trong dãy các phép toán không chứa ngoặc thì thực hiện từ trái sang phải, theo thứ tự các phép toán nhân (*), chia (/), chia nguyên (div), lấy phần dư (mod) thực hiện trước và các phép toán cộng (+), trừ (-), thực hiện sau. **Câu 5**: Trong ngôn ngữ lập trình **Pascal**, biểu thức số học $\mathbf{a^2}$ được viết như sau: A. a**.**a B. a × a C. sqrt(a) D. a*****a hoặc sqr(a) **Câu 6**: Biểu thức 17 div 3 + 17 mod 3 có kết quả là: A. 7 B. 8 C. 9 D. 10 **Câu 7**: Trong ngôn ngữ lập trình **Pascal**, Biểu thức lôgic **-1 ≤u ≤ 0** được viết như sau: A. -1 ≤ u or u ≤ 0 B. -1 <= u or u <= 0 C. -1 ≤ u and u ≤ 0 D. -1 <= u and u <= 0 **Câu 8**: Trong ngôn ngữ lập trình **Pascal**, Biểu thức toán học $\frac{-b+\sqrt{b^2-4ac}}{2a}$ được viết như sau: A. (-b+sqr(b*b-4*a*c)) / (2*a) B. [-b+sqr(b*b-4*a*c)] / (2*a) C. (-b+sqrt(b*b-4*a*c)) / (2*a) D. [-b+sqrt(b*b-4*a*c)] / (2*a) **Câu 9**: Biểu thức nào sau đây **ĐÚNG** khi cả A và B đều **ĐÚNG**? A. A + B B. A and B C. not(A or B) D. not(A) or not(B)

NỘI DUNG	GHI CHÚ

	Câu 10: Xét biểu thức lôgic: (**m mod 100 < 10**) **and** (**m div 100 > 1**), với giá trị nào của m dưới đây biểu thức trên cho giá trị **TRUE**.
	A. 63 B. 105 C. 207 D. 311
	Câu 11: Trong **Pascal**, câu lệnh gán có dạng:
	A. <tên biến> **:=** <biểu thức>; B. <tên biến> **: =** <biểu thức>;
	C. <tên biến> **=:** <biểu thức>; D. <tên biến> **= :** <biểu thức>;
	Câu 12: Trong ngôn ngữ lập trình **Pascal**, để thực hiện phép gán giá trị biểu thức (**a + b**) cho biến **S**, ta viết:
	A. a+b := S; B. S = a + b; C. S =: a + b; D. S := a + b;
	Câu 13: Phép gán giá trị toán học $S = \sqrt{p(p-a)(p-b)(p-c)}$ được viết trong Pascal như sau:
	A. S := sqrt(p(p-a)(p-b)(p-c)); B. S = sqrt(p*(p-a)*(p-b)*(p-c));
	C. S := sqr(p*(p-a)*(p-b)*(p-c)); D. S := sqrt(p*(p-a)*(p-b)*(p-c));
	Câu 14: Cho khai báo biến:
	`var m, n: integer;`
	`x , y: real;`
	Lệnh gán nào sau đây **KHÔNG** thực hiện được:
	A. m := -4; B. n := 3.5; C. x := 6; D. y := +10.5;

1. THÔNG TIN CỦA HỌC SINH:

Trường THPT Nguyễn Tất Thành

Lớp: 1…A…

Họ tên học sinh: .....

2. THẮC MẮC CỦA HỌC SINH:

NỘI DUNG HỌC TẬP	CÂU HỎI CỦA HỌC SINH	GIÁO VIÊN GIẢI ĐÁP
TUẦN 6		

3. THÔNG TIN CỦA GIÁO VIÊN:

Giáo viên hỗ trợ: **NGUYỄN THÀNH TÀI**

Số điện thoại của giáo viên: **0917036462** (Zalo)

Email: **taint74@gmail.com**