



# C言語

菊地

詫び。



# お品書き

- 前回までのおさらい + 演習問題!  
(聞きながらpaiza立ち上げておいてねー！)
- **きちんとした配列。**  
(次回前半をifなどの条件文、配列とアドレスの関係は次回の後半に回したいと思います！)
- 今までのおさらいと演習問題！

# 前回までの おさらい

- **第一回目**

->出力関数printf, 繰り返し構文(for, while, do...while)

- **第二回**

->演算子、型、printfの使い方！

->ちょっとだけ配列...(char \*って何だろうはもうちょい後に回します！)

# 第一回

---

**printf**

---

**for文**

---

**While文**

---

**do···while文**

# 第一回

---

printf

←標準出力関数、  
第一引数を出力

---

for文 ←繰り返す回数がわかっている  
ときに使う

---

While文 ←繰り返す回数がわからない  
ときに使う

---

do…while文 ←1回以上繰り返したいと  
きに使う



# やってみよう！

- paiza.ioというサイト (<https://paiza.io/ja>)で試しに書いてみよう！

3.これが出れば正解！



1. コード作成を試してみるを押す。



2.左上の丸井ボタンの矢印を押してCを選択



# 復習問題！

- **Q1.** 30を7で割ったあまりの回数だけHi!って表示してみよう！

ヒント: 余りは余剰、つまり前回の演算子…%

(制限時間3分)



# 解答！

---

- **Q1.** 30を7で割ったあまりの回数だけHi!って表示してみよう！

ヒント: 余りは余剰、つまり前回の演算子…%

```
1  /*参照するライブラリを指定*/
2  #include <stdio.h>
3
4  /*戻り値の型 関数名 (引数の型)*/
5  int main(void)
6  {
7      int i;
8
9      /*30を7で割ったあまり、つまり30%-7=-2*/
10     /*2回出力されるよ!*/
11     while (i < (30%-7))
12     {
13         printf("Hi");
14         i++;
15     }
16     /*main関数の戻り値*/
17     return 0;
18 }
```

# 復習問題！

- **Q2.** 1,2,3,⋯10までを表示してみよう！

ヒント:繰り返し用の変数は処理の中でも使えるよ！たとえば、

```
int i;
```

```
for(i = 0;i<最大値;i++)
```

```
{printf(“%d”, ここにも使える );⋯.}
```

おっとこれ以上は言えねえ…

(制限時間2分)

# 解答！

- **Q2.** 1,2,3,...10までを表示してみよう！

ヒント:繰り返し用の変数は処理の中でも使えるよ！たとえば、

```
int i;
```

```
for(i = 0;i<n;i++)
```

```
{printf("%d", ここにも使える );...}
```

おっとこれ以上は言えねえ…

```
1  /*参照するライブラリを指定*/
2  #include <stdio.h>
3
4  /*戻り値の型 関数名 (引数の型)*/
5  int main(void)
6  {
7      int i;
8
9      for (i = 0; i < 10; i++)
10     {
11         printf("%d", i + 1);
12     }
13     /*main関数の戻り値*/
14     return 0;
15 }
```

# 第二回

## 間違えやすい 演算子まとめ

演算子	意味
$x=y$	xにyを代入
$x==y$	xとyは等しいか？
$x>=y$	xはy以上
$x(\text{算術演算子})=y$ $x(\text{算術演算子})y$ の結果をxに代入	

# インクリメント・デクリメント演算子

演算子	意味
++X	?
X++	?
--X	?
X--	?

# インクリメント・ デクリメント演算子

演算子	意味
$++X$	(前置演算)xに <b>1足してxを評価</b>
$X++$	(後置演算)xを <b>評価してからxに1足す</b>
$--X$	(前置演算)xから <b>1引いてxを評価</b>
$X--$	(後置演算) xを <b>評価してからxから1引く</b>

# 型の種類

実行環境によって異なる

型名	意味	最大値	大きさ
void	型なし	-	-
char	文字型	-128 ~ 128	1
int	整数型	-2147483648~2147483647	4
float	単精度浮動小数点型	-3.402823e+38 ~ 3.402823e+38	4
double	倍精度浮動小数点型	-1.797693e+308~1.797693e+308	8

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i;
6
7      i = 0;
8
9      /*選択肢1*/
10     printf("Output-1\n");
11     while (i++ < 5)
12     {
13         printf("Hi?");
14     }
15
16     /*選択肢2*/
17     printf("Output-2\n");
18     while (++i < 5)
19     {
20         printf("Hi!");
21     }
22 }
```

Q4.

以下の出力はどちらが多い  
かな？



# こたえ！

---

- 以下の出力はどちらが多いかな？
- 選択肢1のほうが多いよー！
- 0からはじまるのと、1から始まるのと

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i;
6
7      i = 0;
8
9      /*選択肢1*/
10     printf("Output-1\n");
11     while (i++ < 5)
12     {
13         printf("Hi?");
14     }
15     i = 0;
16
17     /*選択肢2*/
18     printf("Output-2\n");
19     while (++i < 5)
20     {
21         printf("Hi!");
22     }
23 }
```

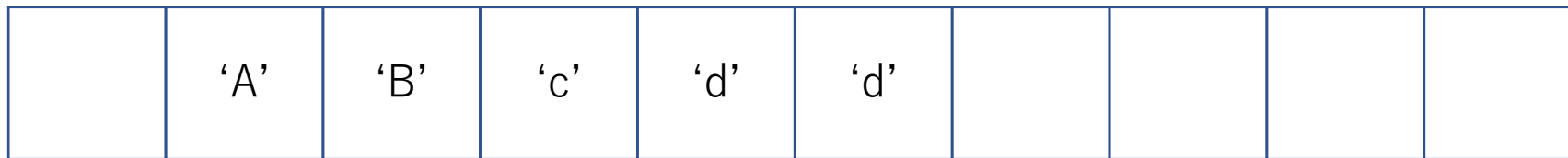
本題!

# 配列

- 定義の仕方と使い方、注意点。
- 演習問題



配列名  
=A[5]



0番目 1番目 2番目 3番目 4番目

# 定義の仕方と使い方

## ・定義

型名 配列名[配列の要素の数]

## ・要素の取り出し方

配列名[要素の先頭からの番号]

## ・注意点

添え字の始まりは0から。最後は要素の**最大値-1**までしかない。

**初期値は不定！  
自分で入れないといけない！**

```
1  /*参照するライブラリを指定*/
2  #include <stdio.h>
3
4  /*戻り値の型 関数名 (引数の型)*/
5  int main(void)
6  {
7      /*宣言*/
8      /*型 配列名 [配列の大きさ]*/
9      int a[5];
10     char b[20];
11
12     /*要素の入れ方*/
13     /*配列名[添え字] = 入れたい値。*/
14     a[0] = 10;
15     b[3] = 'c';
16
17     /*初期値であれば中カッコでいっぺんに入れることもできる。*/
18     int c[3] = {5, 1, 3};
19     /*この時は、c[0] = 5, c[1] = 1, c[2] = 3*/
20     /*いっぺんに初期化することも可能*/
21     int d[4] = {0};
22
23     /*繰り返し構文用の添え字*/
24     int i;
25     int ans;
26     /*この書き方であればiは10未満、つまり9まで動く*/
27     for (i = 0; i < 5; i++)
28     {
29         a[i] = i;
30     }
31     /*main関数の戻り値*/
32     return 0;
```

# 配列に入れた値の確認と計算

- for文やwhile文と一緒に組み合わせることができる！
- ヒント: for文の中でifと組み合わせれば、偶数や奇数のみの足し算もできるよ！

```
1  /*参照するライブラリを指定*/
2  #include <stdio.h>
3
4  /*戻り値の型 関数名 (引数の型)*/
5  int main(void)
6  {
7      /*宣言*/
8      /*型 配列名 [配列の大きさ]*/
9      int a[5];
10
11     /*繰り返し構文用の添え字*/
12     int i;
13     int ans;
14     /*この書き方であればiは10未満、つまり9まで動く*/
15     for (i = 0; i < 5; i++)
16     {
17         a[i] = i;
18     }
19     /*解答格納用の変数を宣言*/
20     int ans;
21     /*変数を初期化、しないとゴミ(変な値)が入っている*/
22     ans = 0;
23     /*繰り返し構文と計算の組み合わせ*/
24     for (i = 0; i < 5; i++)
25     {
26         /*配列の値の確認*/
27         printf("a[%d] = %d\n", i, a[i]);
28         /*ansにa[i]を足していく*/
29         ans += a[i];
30     }
31     /*結果の表示！*/
32     printf("%d\n", ans);
33
34     /*main関数の戻り値*/
35     return 0;
```

# 演習問題！

- Q6.

**31から37**まで1間隔の整数値が入った配列を作って、  
**先頭から2番目の要素**と**先頭から5番目の要素**を足した結果を表示してみよう！

- ヒント：for文は初期値と終了の値を指定できたよ…？配列も値の場所を決められたはずだ…

(所要時間は10分)

# 演習問題！

---

- Q6.

31から37まで1間隔の整数値が入った配列を作って、2番目の要素と5番目の要素を足した結果を表示してみよう！

- ヒント：for文は初期値と終了の値を指定できたよ…？配列も値の場所を決められたはずだ…

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int a[7];
6      int i;
7
8      /*解答1*/
9      for (i = 31; i < 38; i++)
10         a[i - 31] = i;
11
12     printf("answer = %d\n", a[1] + a[4]);
13
14     i = 31;
15     /*解答2*/
16     while (i < 38)
17     {
18         a[i - 31] = i;
19         i++;
20     }
21     printf("answer = %d\n", a[1] + a[4]);
22 }
```

# 演習問題！

- Q5.1から100までの平均を出して、それを表示してみよう！  
ヒント：割り算の結果は少数部が…あれ？整数型ではない…？  
表示の仕方は…？  
おっと、これ以上は人が来たようd(ry  
制限時間(7分)



# 解答！

---

- Q5.1から100までの平均を出して、それを表示してみよう！

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      double ans;
6      int i;
7
8      ans = 0;
9      i = 0;
10     /*i = 1; からにしてwhile(i <= 100)でもいいよ！*/
11     while (i < 100)
12     {
13         ans += (i + 1);
14         i++;
15     }
16     /*答えを違う変数に格納してもいいよ！*/
17     ans /= 100;
18     printf("average = %f\n", ans);
19 }
```

# 次回！条件文について！

- **if**
- **if... else if... else**
- **switch**

# アンケートへの解答

- 競技プログラミングについて

->ライン(おいでよU-labの森2021)に参加している人は5/16のラインを参考に！

- c言語を使ったアプリの作り方！

->ライブラリ関数の自作をやってみるといいかも！手始めに、strlen関数を自作してみよう！（現段階でstrchr関数とかも自作できるはず。）

# 告知

- 今回もアンケートをコメント欄に貼っておきました！答えてくれたらうれしーなー！

- UU-Circleについてのアンケートにお答えください！

([https://docs.google.com/forms/d/e/1FAIpQLScm6ocHadaIWp8PnXh7Cfw2f5upkHTFP-5yoh\\_vdMCNJ9N5LQ/viewform](https://docs.google.com/forms/d/e/1FAIpQLScm6ocHadaIWp8PnXh7Cfw2f5upkHTFP-5yoh_vdMCNJ9N5LQ/viewform))

---

FIN!

