```
//以下をコピペしてご利用ください
//コードA｜プログラム「GetYahooAuctionData」
function  GetYahooAuctionData(){

 //コードA1｜スプレッドシートのシート名を取得
 var sheetname = 'シート1';

 //コードA2｜スプレッドシートの設定
 var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheetname);

 //コードA3｜最終行の取得
 var lastrow = sheet.getLastRow();

 //コードA4｜8行目以下のデータを削除
 sheet.getRange(8,1, lastrow-8, 7).clearContent();

 //コードA5｜product_listという配列を作成
 var product_list = [];

 //コードA6｜セルB1の検索キーワードを取得
 var keyword = sheet.getRange('B1').getValue();

 //コードA7｜YahooオークションのURLを取得(1~100件の商品情報を取得)
 var url = 'https://auctions.yahoo.co.jp/search/search?p=' + keyword + '&n=100';

 //コードA8｜url(コードA7)で指定したウェブサイト情報を取得
```

```
var html = UrlFetchApp.fetch(url).getContentText();

//コードA9｜検索結果の件数を取得
var items = Parser.data(html)
.from('<div class="Tab__itemInner">')
.to('</div>')
.iterate();

//コードA10｜検索結果の件数の「すべて」と「件」を削除
var item = trimming(items[0]).replace('すべて','').replace('件','');

//コードA11｜検索結果の件数を100で割ったの商（余りを切り捨てた値）を取得
var counter = Math.floor(item/100);

//コードA12｜コードBのプログラムを呼び出す（「html, product_list, 0」の3つを引数とする）
product_list = Pagecrawling(html, product_list, 0);

//コードA13｜k=1,2,・・・,counter(コードA11)で繰り返す
for(var k=1; k <= counter; k++){

    //コードA13-1｜pageurlのウェブページを取得(k＝1で101~200件の商品情報、k＝2で201~300件の商品情報を取得)
    var pageurl = 'https://auctions.yahoo.co.jp/search/search?p=' + keyword + '&b=' + k + '01&n=100';

    //コードA13-2｜コードA13-1で指定したURLのウェブサイト情報を取得
    html = UrlFetchApp.fetch(pageurl).getContentText();
```

```
        //コードA13-3｜コードBのプログラムを呼び出す（「html, product_list, 0」の3つを引数とする）
        product_list = Pagecrawling(html, product_list,k);
  }

  //コードA14｜product_listに格納された配列の値を8行目以下に貼り付け
  sheet.getRange(8,1, product_list.length, product_list[0].length).setValues(product_list);

  //コードA15｜「商品件数」と「item」(コードA11で取得)を文字列で結合
  var report1 = '商品件数:' + item;

  //コードA16｜セルB3にrepot1(コードA16)を貼り付け
  sheet.getRange('B3').setValue(report1);

  //コードA17｜「検索日時」と「今の日時」を文字列で結合
  var report2 = '検索日時:' + Utilities.formatDate(new Date(), 'Asia/Tokyo', 'yyyy/MM/dd hh:mm:ss');

  //コードA18｜セルB4にrepot2(コードA18)を貼り付け
  sheet.getRange('B4').setValue(report2);
}

//コードB｜プログラム「Pagecrawling」(「html, product_list, k」の3つの引数を受け取る)
function Pagecrawling(html, product_list, k){

  //コードB1｜kに100を掛ける（kensuは商品No取得時に使用）
```

```
var kensu = k * 100;


//コードB2｜Yahooオークションサイトのhtml情報から「商品ごとの情報を取得」
var products = Parser.data(html)
.from('<li class="Product">')
.to('</li>')
.iterate();


//コードB3｜i=0,1,・・・,「products.length」で繰り返す（Yahooオークションの各商品情報を解析）
for(var i=0; i<products.length; i++){


        //コードB3-1｜各商品の件名情報を取得
        var titles = Parser.data(products[i])
  .from('<h3 class="Product__title">')
  .to('</h3>')
        .iterate();


        //コードB3-2｜コードCのプログラムを呼び出す（titles[0]が引数）→各商品の件名情報のうち、余計な記述を削除
        var title = trimming(titles[0]);


        //コードB3-3｜各商品の商品URLを取得
        var urls = Parser.data(products[i])
        .from('<a class="Product__titleLink" href="')
        .to('" target="_blank" rel="noopener noreferrer">')
  .to('</div>')
        .iterate();
```

```
//コードB3-6｜fixedprice(即決価格)の変数を設定し、「-」を格納

var fixedprice = '-';


//コードB3-7｜currentprice(現在の価格)の変数を設定し、「-」を格納

var currentprice = '-';


//コードB3-8｜コードCのプログラムを呼び出す（prices[0]が引数）→各商品の価格情報のうち、余計な記述を削除

var price = trimming(prices[0]);


//コードB3-9｜価格情報を「円」で区切り、配列化する

var myprice = price.split('円');


//コードB3-10｜j=0,1,・・・,「myprices.length - 1」で繰り返す（各商品の価格情報を調査）

for(var j=0; j<myprice.length-1; j++){


//コードB3-10-1｜「myprice[j]」に「即決」という文字列が含まれているならば（即決価格であれば）

if (myprice[j].indexOf('即決') !== -1) {


//コードB3-10-2｜fixedpriceにmyprice[j]を格納

fixedprice = myprice[j];


//コードB3-10-3｜「myprice[j]」に「現在」という文字列が含まれているならば（現在の価格であれば）

}else if(myprice[j].indexOf('現在') !== -1){
```

```
        //コードB3-10-4｜currentpriceにmyprice[j]を格納
        currentprice = myprice[j];
        }
        }

        //コードB3-11｜各商品の入札情報を取得
        var labels = Parser.data(products[i])
   .from('<div class="u-displayIB u-marginR15">')
   .to('</div>')
        .iterate();

        //コードB3-12｜コードCのプログラムを呼び出す（labels[0]を引数とする）→各商品の入札情報のうち、余計な記述を削除
        var label = trimming(labels[0]);

        //コードB3-13｜各商品の残り時間を取得
        var times = Parser.data(products[i])
        .from('<div class="u-displayIB">')
        .to('</div>')
   .iterate();

        //コードB3-14｜コードCのプログラムを呼び出す（times[0]を引数とする）→各商品の残り時間情報のうち、余計な記述を削除
        var time = trimming(times[0]);

        //コードB3-15｜product_listの配列末尾に、コードB4内で取得した各データを格納
        product_list.push([kensu + i+1,title,url, currentprice,fixedprice,label,time]);
```

```
  }

  //コードB4｜product_listの情報をコードAに返す
  return product_list
}

//コードC｜プログラム「trimming」(「target」の1つの引数を受け取る)
function trimming(target){

  //コードC1｜引数として受け取ったtarget情報からタグ情報を削除
  var seikei1 = target.replace(/<("[^"]*"|'[^']*'|[^'">])*>/g, '');

  //コードC2｜seikei1(コードC1)の情報から余白を削除
  var seikei2 = seikei1.replace(/\s+/g,'');

  //コードC3｜seikei2(コードC2)の情報をコードBに返す
  return seikei2;
}
```