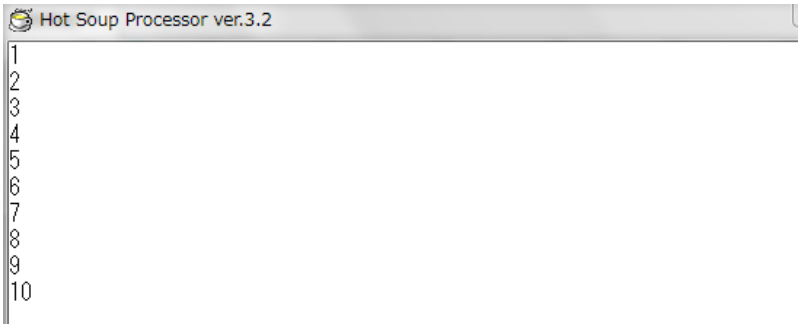
０．はじめに

今回は、プログラムに同じ処理を繰り返させる、「ループ」について、説明します。今回は「repeat ループ」「ラベルを用いたループ」の２つを紹介します。（もう一つ、foreach ループがあるのですが、foreach ループは配列やモジュールの概念について説明するときに触れるため、今は省略します）ループでは、これまで以上に「プログラムは（基本的に）上の行から順番に処理されていく」ことを念頭に置いて、処理の順番を自分で追っていかなければなりませんし、また、変数と同じくかなり「プログラム的な考え方」をするため、若干これまでより内容が難しく感じるかもしれません。しかし、HSP でのプログラムに限らず、プログラミングの基礎は「ループ」と「分岐」であるため、かなり重要な内容で、今後必ず必要となります。今回でマスターするよう頑張りましょう。

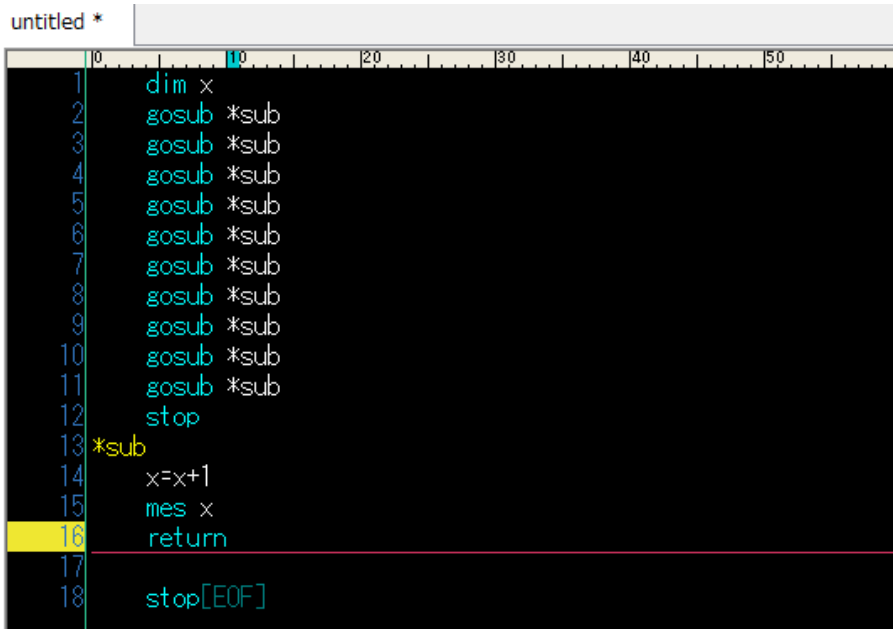
１．repeat ループ

repeat ループは、repeat 命令を用いたループのことです。プログラム中で何度も同じ処理を繰り得させる場合、これを使用します。実際にプログラムを見て考えてみましょう。

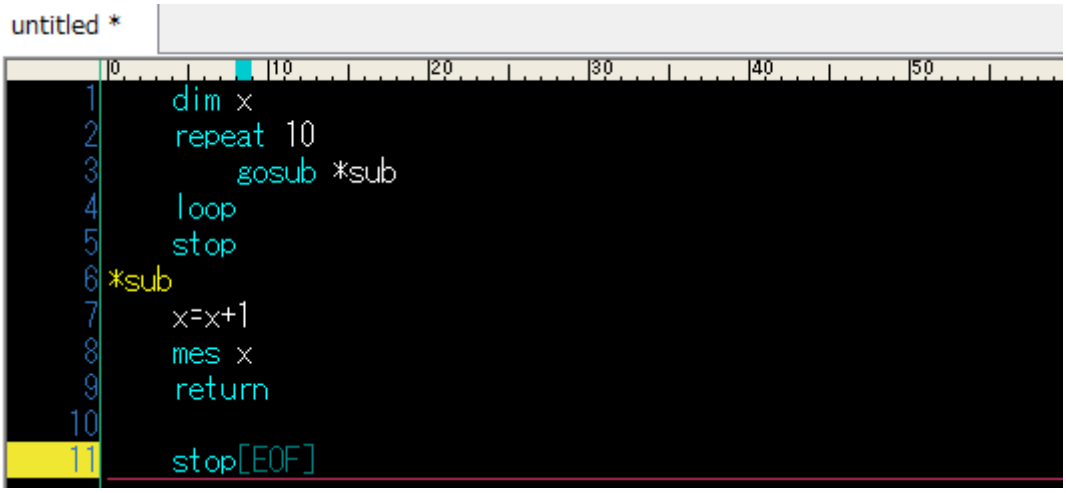
さて、前回このような実行結果を表示させる課題に挑戦したのは覚えているでしょうか？



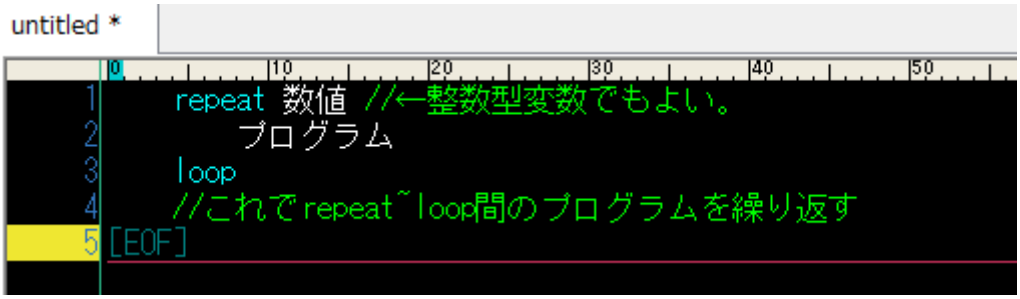
これを表示させるのに、下のようなプログラムになったのではないでしょうか。



このプログラムを見ると、2 行目から 12 行目にかけて、まったく同じ命令が 10 回使われています。もしも、「gosub 命令を 10 回繰り返させる」という命令があれば、とてもプログラムがすっきりしますね。そこで登場するのがループです。では、repeat ループを使って上のプログラムを書き直してみます。



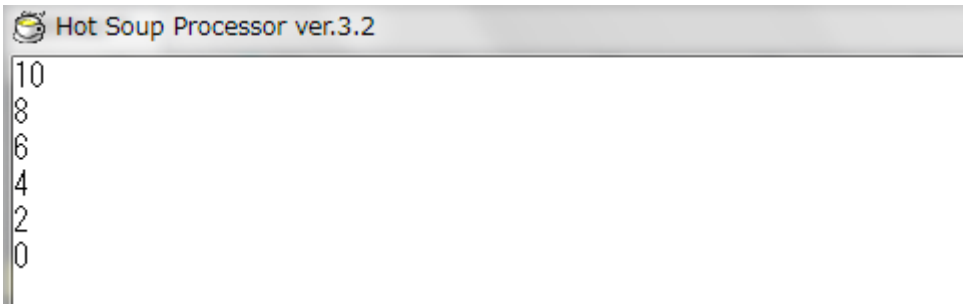
どうでしょうか、さっきと比べてずいぶんすっきりしたと思います。実際に実行して、同じ結果になることを確認してみてください。repeat 命令は、次のような形で使います。



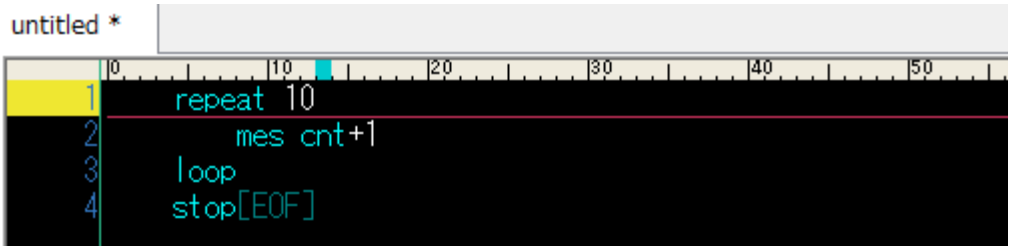
上の図で示してある通り、repeat~loop 間のプログラムを repeat の後の数値（直接数字を入れてもいいし、変数でもいい）の数だけ繰り返します。中の繰り返すプログラムは、1 行でなくとも、何行でも大丈夫です。（プログラムの追い方としては、上から順番に処理していき、repeat の行は読み飛ばし、loop 命令に到達した段階で、今現在何回目のループかを確認し、指定回数終了していなければ repeat 命令の行まで戻るといった感じ）しかし、1 つだけ守らなくてもエラーが出るわけではないですが、守っておいたほうがいい決まりがあります。repeat～loop 間のプログラムは tab 一つ分ずらすようにしてください。これによって、どこまでがループなのかがわかりやすくなります。ラベルとプログラムとの区別をつけるために tab をあけるのと同じく、これも守るようにしてください。後々長めのプログラムを書くときに、バグの発見や後から見直す際に、こういったことがとても重要になってくるので、くせにするようにしてください。

課題１．１

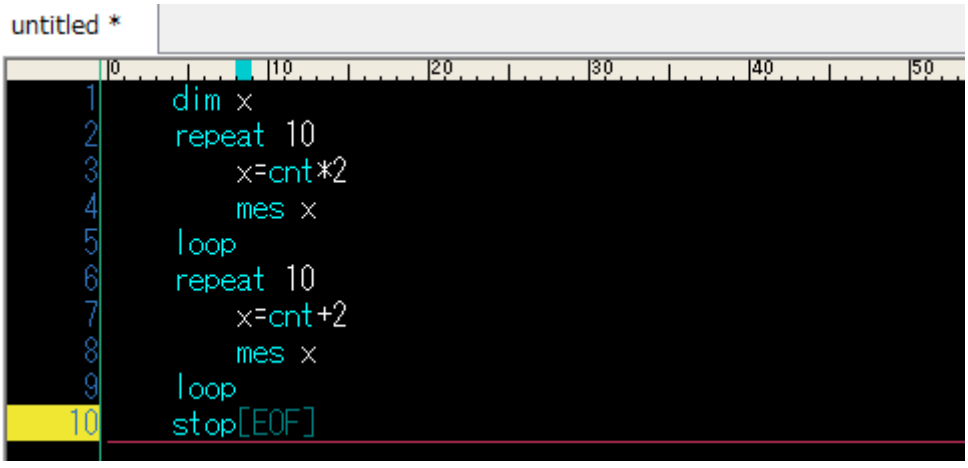
ループを使って、変数 x を 10 から 0 まで 2 ずつ減少させ、表示させるプログラムを作りなさい。なお、サブルーチンは使ってはならず、使える変数は 1 個まで、mes 命令は 1 か所でのみ使用できる。実行結果の例を以下に示す。



さて、repeat ループには、とても便利な補助機能があります。実はそれを使うことで先ほどのプログラムはもっと簡略化できるので、それを確認しておきましょう。



これだけ？というかんじですが、この 4 行でさっきとまったく同じ実行結果が得られます。実際に実行してみてください。さて、mes 命令で、見慣れぬ cnt という命令が出てきています。これが「便利な機能」なのですが、この cnt は別名「ループカウンタ」と呼ばれて、現在がループの何回目かを記録してくれる変数です。水色なのに変数です。ちょっと紛らわしいので注意してください。cnt は整数値型変数なので、ほかの変数と同じように足し算や掛け算など、計算に使うことができますが、値を代入することはできないので注意してください。また、ループごとに別々の cnt を持っていることに注意してください。

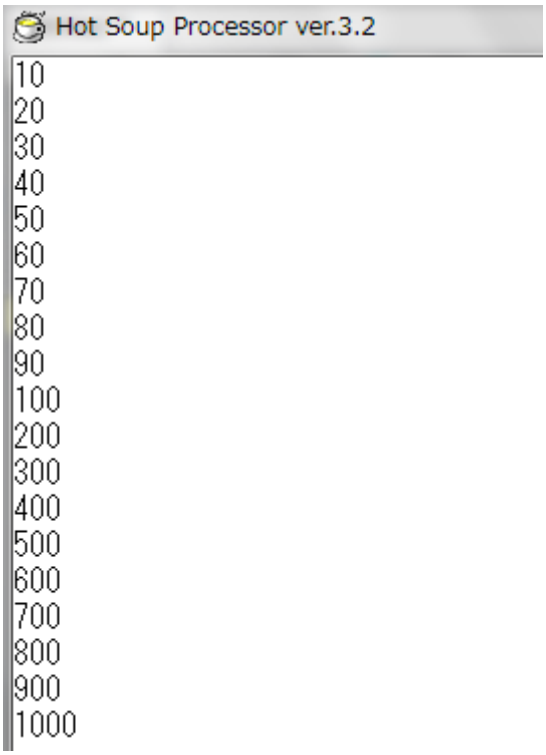


上のプログラムで、2 つループが出てきています。どちらのループでも変数 x は共通のも

のですが、変数 cnt は、それぞれのループで名前は同じでも要素は全く別の cnt を持っています。この考えは今回の後半で出てくる「2 重ループ」において重要になるので、頭の片隅に入れておいてください。

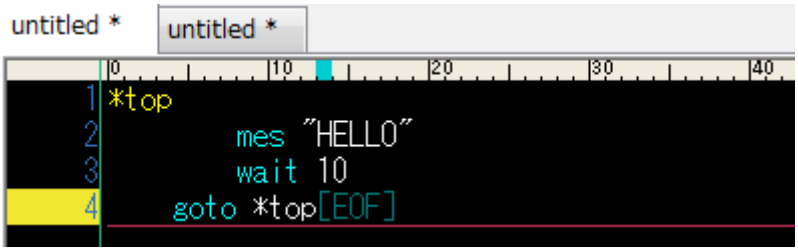
課題１． ２

次のような結果を表示するプログラムを作れ。ただしループカウンタ cnt 以外に変数を使用してはならず、mes 命令が使えるのは 2 か所までとする。ヒント…数式と同じように、変数同士の計算でも×と÷は＋やーと比べて計算の優先度が高く、＋やーを優先させるには（）で囲む必要がある。



２．ラベルによるループ

さて、もう一つ、前回やったラベルを使ってループを作ってみます。gosub 命令とラベルの使い方は覚えているでしょうか？実際のプログラムを見ながら思い出してみてください。ただし、今回のプログラムは動作が停止する可能性があるので、できれば実行しないでください。



プログラムを順に追っていきましょう。まず、1 行目のラベルはただの目印でプログラムの処理とは関係ないので読み飛ばします。次に mes 命令で「HELLO」と表示します。ここまでは大丈夫です。次に新しく出てきた「wait」ですが、これは一定時間待つ（wait＝待つ）という命令で、命令のあとで指定した数（整数型変数でもよい）×10m s（100 分の 1 秒）だけ何もせずに待機させることが出来ます。今回は 10×10m s ＝0.1 s （s は second の s で秒を意味する単位）待機します。そして次の行に行き、goto 命令で最初の行のラベルに戻り…と永遠と同じ処理を繰り返します。これのように永遠続くループのことを無限ループと呼んで、無限ループを作る場合は、必ず wait 命令を入れるか、終了条件を入れる（これについては次回の内容である分岐で触れます。）ようにしてください。今回はまだ終了条件について習っていないので wait 命令で動作を遅くさせています。ラベルによるループは面倒な部分が多く、repeat 命令と違いループカウンタで cnt を使えないといった不便さはありますが、repeat ループには repeat-loop 間にラベルを入れると動作が不安定になるという弱点があるため、場合によってどちらかを使い分ける必要が出てきます。これについては、第５回で触れるので、今は「無限ループならラベル、有限ループなら repeat」というように覚えてくれて大丈夫です。

課題２． 1

実行直後から永遠と x が増え続け、それを表示させるプログラムを作りなさい。wait 時間は１ s として、実行時は動作が確認でき次第ウィンドウを閉じること。(エラー防止のため)

３．ループを使って様々な処理

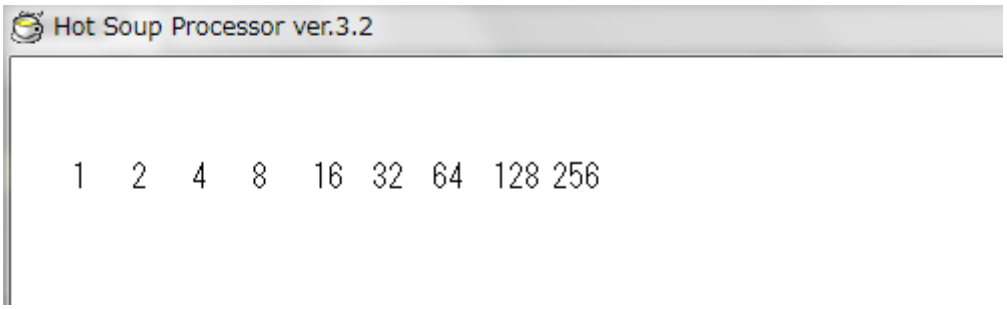
ループには様々な活用法があります。課題でいろいろな使い方を確認してみましょう。これは、プログラム的な考え方の練習にもなります。また、前回軽く触れた命令や関数をいくつか使っているほか、新しく出てくる関数・命令もあります。これらの使い方を見るときはヘルプを見るといいでしょう。今後自分でプログラムを書く場合はとても大切なので、今のうちにやり方をさっと確認しておきましょう。プログラムを書いている画面でＦ１キーを押すか、上のツールバーのヘルプからＨＳＰキーワード検索を選択してください。

この画面左上のボックスに命令や関数名を打ち込むことでそれらについて調べることが

できます。実際に mes 命令など、いままで取り扱ったものを調べてみるといいでしょう。

課題３．１

ループを使って変数 x を 1 から 256（2 の 8 乗）まで 2 倍でどんどん変化させ、その値を表示するプログラムを作りなさい。なお、使える変数は x の 1 つまでとし、mes 命令は 1 か所のみ使用でき、座標を変数式で示した pos 命令によって横向きに並べること。実行例を下に示す。pos 命令の使い方は F1 キーを押してヘルプにより確認するか、前回の HSP 講座 2 を確認しなさい

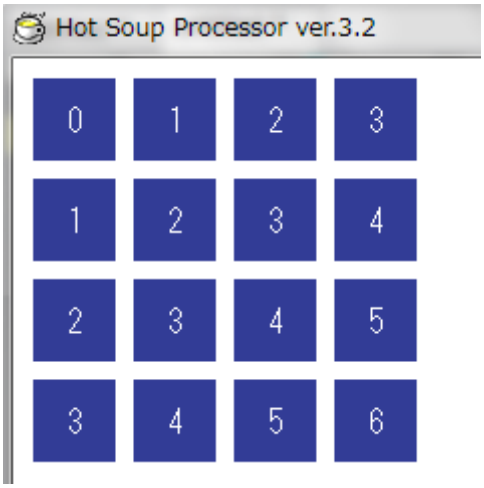


課題３．２

ループを用いてグラデーションを作りなさい。なお、color 命令で色を指定し、座標を変数式で指定した boxf 命令で四角形を描くことによって実現すること。それぞれの命令の使い方は F1 キーを押してヘルプにより確認するか、前回の HSP 講座 2 を確認しなさい。実行例を以下に示す。同じような結果が得られれば、グラデーションは 1 つのパターンだけで四角形の大きさや色が違っていてもよい。（なお、終着色が白や黒でないので右側のようなグラデーションを作るほうが若干難しいかもしれない）

課題３．３（難？）

次のような表示をするプログラムを作りなさい。なお、boxf 命令および mes 命令は１か所でのみ使用することができ、ループの中にさらにループを入れる（2 重ループと呼ぶ）ことで実現せよ。（具体的には横に四角形と文字を表示させるループ（課題３．１のような処理）を、縦に繰り返させるループにすればいい。もちろん縦横どちらが先でも構わない）この際､内側のループと外側のループでは異なる cnt を持っているということに十分注意すること｡実際には変数を用意し､外側の cnt を用意した変数に代入することで内側のループでその値を参照すればよい。また、文字色や塗りつぶした四角形の色は、文字と四角の両方が目視できれば実行例と異なっていてもよい。四角形を描くためには boxf 命令、位置表示には pos 命令、色指定には color 命令がある。それぞれの命令の使い方は F1 キーを押してヘルプにより確認するか、前回の HSP 講座 2 を確認しなさい。



課題３．４

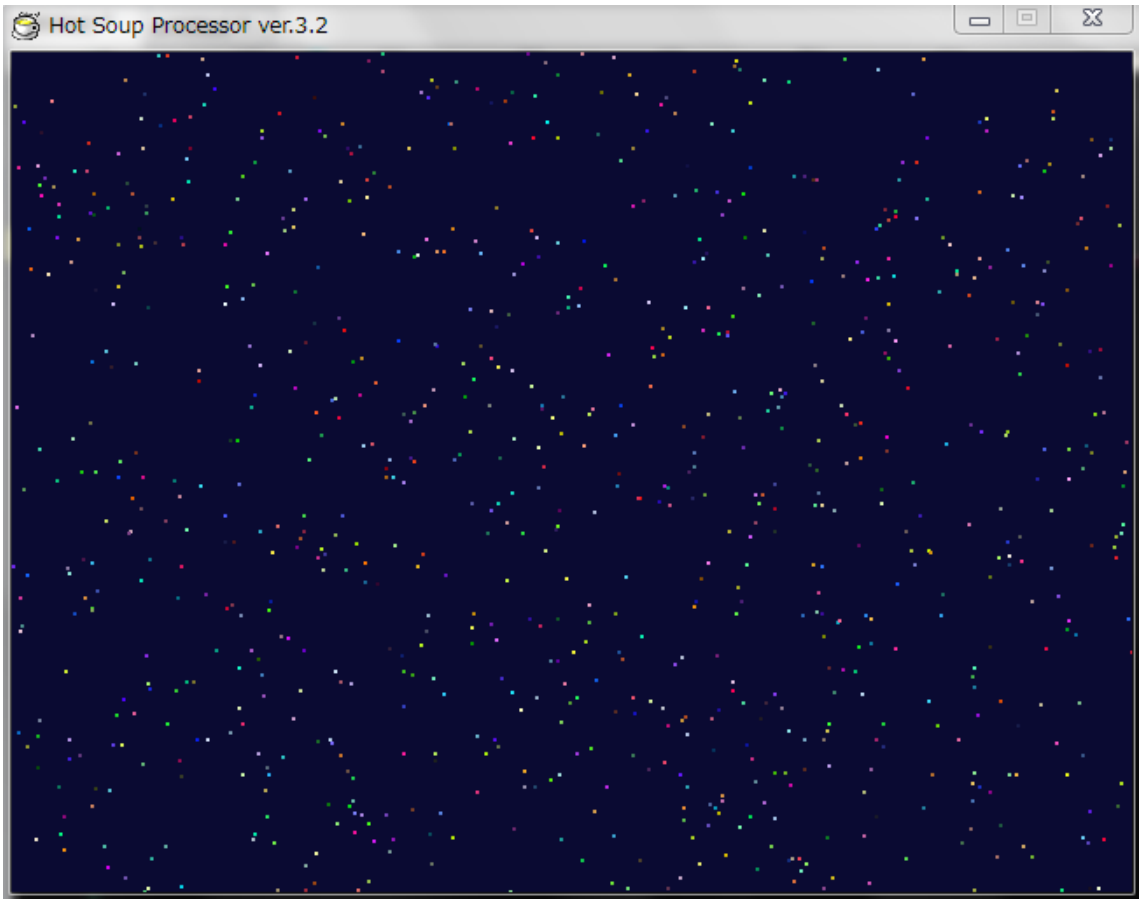
ループを用いて、現在の時刻を常に表示させ続ける時計を作成しなさい。表示内容は、月、日、時、分、秒の５つで、１行目に月日を、２行目に時と分と秒を表示させなさい。時間を読み取る関数は gettime、表示位置の指定に pos 命令、前に表示させた内容を消すのに boxf 命令、塗りつぶし色と文字色の指定に color 命令を使うことができる。処理順番としては、前の表示の塗りつぶし→現在の時刻表示の順で行わせ、これを無限ループにより永遠繰り返させればよい。待機時間（wait）は 100ms とする。それぞれの命令の使い方は F1 キーを押してヘルプにより確認するか、前回の HSP 講座 2 を確認しなさい。

課題３．５

ループを用いて現在のウィンドウ上のマウス座標を常に表示するプログラムを作りなさい。マウスの座標は関数 mousex および mousey により読み取れる。それぞれの使い方はＦ１キーを押し、自分で調べなさい。ほかの手順は課題３．４とおおよそ同じである。

課題３． ６

ランダムな値を返す関数 rnd および初期化命令 randomize を使い、下のような夜の空をイメージした画像を作りなさい。画面を濃い群青色で塗りつぶした後、ランダムな位置にランダムな色の点を boxf 命令で打つことで描いている。描く点の数は任せるが、多すぎても少なすぎるのもよくない。自分で適当に調整しなさい。



今回、ループを使うことでいろいろなことをすることができました。ループは次回にやる分岐と組み合わせることでより強力になります。ループ＋分岐で何ができるか。それは第 5 回（次の次）で様々なプログラムを書くことで確認できます。ループは今後もどんどん使っていくので今のうちにしっかり確認しておきましょう。