

① ついに始まった小学校 プログラミング教育 —その現状と課題—



阿部和広 | 青山学院大学大学院 社会情報学研究科

2020年4月、プログラミング教育を含む新しい小学校学習指導要領が実施された。しかし、新型コロナウイルスによる非常事態宣言の発令により、大部分の小学校で休校が続いており(2020年5月24日現在)、教員はその対応で手一杯になっている。その結果、プログラミングの授業は後回しになっているように見受けられる。筆者が行う予定だった教員研修や授業もキャンセルされ、研究授業の講師依頼もほぼなくなった。図らずも、小学校にとってプログラミング教育が「不要不急」だったことが明らかになったと言えるかもしれない。

ここでは、小学校のプログラミング教育の過去を振り返るとともに、現状を概観し、その課題と解決法を提案する。

近年のプログラミング教育

2005年6月、GUIやオブジェクト指向プログラミングなどの研究で知られるAlan Kayが東京都杉並区立和田小学校を訪問した。当時の和田小学校では、日本HPが寄贈したノートパソコンが普通教室で使えるようになっており、プロジェクタや無線LAN環境も整備されていた。そこで行われていたのが、Kayが中心になって開発していたSqueak Etoysというビジュアルプログラミング環境を使った授業である。子どもたちが作成したさまざまなプロジェクトを見たKayは次のようなコメントを残している^{☆1}。

「子どもたちは生まれながらに芸術家である。子どもたちには創造性の高い授業にどっぷり漬かってほしい。ただ、そこからサイエンスを学ぶには、自発的に学ぶことを期待するだけでなく、学習するアイデアを自分たちで作ること、自分の頭で生み出すことが必要。それには大人や周りの子どもたちのガイダンス(指導)が重要になる。そこに導くのが学校の役割だ」

この考え方の背景には、Seymour Papertが1960年代から実践するプログラミングによる数学教育(マstrand、マイクロワールド)がある。しかし、当時、それに気付く人はそれほど多くなかったように思われる。

和田小学校では、その後もプログラミングを教科に取り入れた授業が行われ、たとえば、2008年11月から12月にかけて、5年生、6年生を対象に算数の求積の単元についての研究授業が実施されている¹⁾。

しかし、その後、和田小学校に限らず、小学校でプログラミング教育はほとんど行われなくなる。その背景には、コンピュータを使うことが、ソフトウェアを作ることから、既存のアプリケーションやインターネットの情報を活用することによって変わったことがある。このことは、2008年に実施された旧学習指導要領の総則に、「児童がコンピュータや情報通信ネットワークなどの情報手段に慣れ親しみ、適切に活用する学習活動を充実するとともに、視聴覚教材や教育機器などの教材・教具の適切な活用を図ること」とあることからうかがえる(この部分は1998年実施の総則と変わらない)。

^{☆1} ASCII.jp: “パソコンの父” アラン・ケイ氏が和田小学校を訪問——プログラミング言語“Squeak”でアイデアを具現化する授業を見学、<https://ascii.jp/elem/000/000/348/348472/>

Obama 演説と世界的な流行

潮目が変わったのは、アメリカの起業家である Hadi Partovi と Ali Partovi が、2013 年 1 月に設立した「Code.org」という NPO の設立である。Code.org は、すべての人がプログラミングできるようになることを目的としており、Mark Zuckerberg や Bill Gates など著名人の賛同を得ている。その活動として、「Hour of Code」(1 時間だけでもプログラミングしてみよう) というパズル型のチュートリアルを全世界で行い、2013 年 12 月には、Barack Obama 大統領が国民に呼びかける演説を行うに至った。この中で Obama 大統領は、国の将来のために計算機科学を学ぶことの重要性を説いている^{☆2}。

ほかにも、子どもたちが学ぶ環境として、マサチューセッツ工科大学 (MIT) の Michel Resnick らが開発した Scratch などのビジュアルプログラミング言語環境を使うことで、環境の導入やプログラミング自体の学習のコストが下がったことも、プログラミング教育を支える素地となった。これは、パソコンへのインストールを必要とせず、Web ブラウザ上で動作する Scratch 2.0 が、2013 年 5 月に公開されたことも大きい。Scratch

の統計情報を見ると(図-1)、2013 年から明らかに活動が増えていることが分かる(2020 年 4 月の極端な伸びは新型コロナウイルスの影響と考えられる)。

日本では、内閣の日本経済再生本部のもとで 2013 年 1 月に始まった産業競争力会議で、新経済連盟代表理事の三木谷浩史氏が、「エンジニアの質・量ともにレベルを大幅にアップさせる必要がある」として、「基本的プログラミングの教育を教育課程に入れること」を複数回に渡って提案している(その際、Scratch も例示されている)。そして、2016 年 4 月の第 26 回産業競争力会議において、安倍晋三首相が初等中等教育からプログラミング教育を必修化する方針を表明した^{☆3}。その目的は、「日本の若者が第 4 次産業革命の時代を生き抜き、主導するため」である。

小学校プログラミング教育に向けての議論

この方針を具体化するため、2016 年 4 月から 6 月にかけて、文部科学省で「小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議」が開かれた。その議論の取りまとめの中で、プログラミング教育が定義されている^{☆4}。

^{☆2} President Obama asks America to learn computer science, <https://www.youtube.com/watch?v=6XvmhE1J9PY>

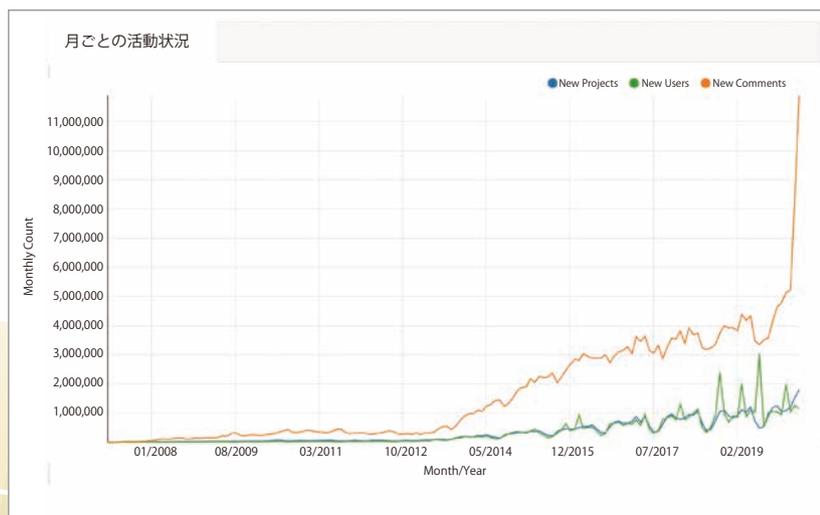


図-1 Scratch の活動状況 (<https://scratch.mit.edu/statistics/>)

「プログラミング教育とは、子どもたちに、コンピュータに意図した処理を行うよう指示することができるということを体験させながら、将来どのような職業に就くとしても、時代を超えて普遍的に求められる力としての「プログラミング的思考」などを育むことであり、コーディングを覚えることが目的ではない」

^{☆3} 日本経済再生本部 産業競争力会議 第 26 回 議事要旨, <https://www.kantei.go.jp/jp/singi/keizaisaisei/skkaigi/dai26/gijiyoushi.pdf>

^{☆4} 小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ), https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/attach/1372525.htm

前述の経緯を見ると、日本のプログラミング教育は、産業界の要請に基づくものであることは明らかである。しかし、学校教育法第18条で定める小学校の目標に職業訓練はない(職業についての基礎的な知識と技能を養うことは、第38条で中学校の目標とされている)。そのために、「時代を超えて普遍的に求められる力」としての「プログラミング的思考」が作られ、「コーディングを覚えることが目的ではない」と念押しされたと考えられる。

プログラミング的思考とは

議論の取りまとめの中で、プログラミング的思考は、以下のように定義されている。

「自分が意図する一連の活動を実現するために、どのような動きの組合せが必要であり、一つ一つの動きに対応した記号を、どのように組み合わせたらいいのか、記号の組合せをどのように改善していけば、より意図した活動に近づくのか、といったことを論理的に考えていく力」

この注釈には、「いわゆる『コンピューショナル・シンキング』の考え方を踏まえつつ、プログラミングと論理的思考との関係を整理しながら提言された定義」とある。コンピューショナル・シンキングとは、提唱者のWingによると、「コンピュータ科学者だけではなく、すべての人が学び、そして使いたいと考えるに違いない一般的な態度とスキルに関するもの」である²⁾。その扱っている対象は広く、プログラミング的思考は、その中の手続き的な考え方を抜き出したサブセットと考えられる。

議論の取りまとめは、「小学校現場の不安感を少しでも軽減」することを狙いの1つとしており、従来の各教科等で育まれる論理的・創造的な思考力がプログラミング的思考を育むとともに、各教科等における思考の論理性も明確にするとしている。合わせて、「アナログ感覚を大事にしていくことの重要性」も挙げられており、これが、後述するコンピュータを使わない「アンプ

ラグド」の考え方につながっていったと考えられる。実際、プログラミング的思考を説明する際は、料理の手順など、コンピュータと関係のない現実の話で例えることが多い。

その後、プログラミング教育の議論は中央教育審議会(中教審)に移り、2016年12月に答申が手交された。その中では、新しい時代に必要となる資質・能力を、「知識・技能」、「思考力・判断力・表現力等」、「学びに向かう力・人間性等」の3つの視点から捉えている。これをプログラミング教育に当てはめると、以下のようになる。

【知識・技能】身近な生活でコンピュータが活用されていることや、問題の解決には必要な手順があることに気付くこと

【思考力・判断力・表現力等】発達の段階に即して、「プログラミング的思考」を育成すること

【学びに向かう力・人間性等】発達の段階に即して、コンピュータの働きを、よりよい人生や社会づくりに生かそうとする態度を涵養^{かんよう}すること

つまり、プログラミング的思考は、資質・能力の一面でしかない。しかし、この中の「プログラミング的思考」が、ことさらに強調されるきらいがある。

新学習指導要領の中の プログラミング教育

中教審答申を受けて、2017年3月に新学習指導要領が告示され、2020年4月から小学校で実施することが示された^{☆5)}。

この総則には、「児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動」を、各教科等の特質に応じて計画的に実施することと書かれている。つまり、プログラミングは独立した教科で

☆5) 小学校学習指導要領(平成29年告示)、https://www.mext.go.jp/content/1413522_001.pdf

はない。その例としては、以下の3つが示されている(図-2)。

算数〔第5学年〕の「B 図形」の(1)における正多角形の作図を行う学習に関連して、正確な繰り返し作業を行う必要があり、更に一部を変えることでいろいろな正多角形を同様に考えることができる場面など」

理科〔第6学年〕の「A 物質・エネルギー」の(4)における電気の性質や働きを利用した道具があることを捉える学習など、与えた条件に応じて動作していることを考察し、さらに条件を変えることにより、動作が変化することについて考える場面」総合的な学習の時間「プログラミングを体験することが、探究的な学習の過程に適切に位置付くようにすること」

これらが教科書検定基準となり、教科書にはこれに基づいた内容が掲載されている。ただし、これ以外の教科や単元で、プログラミングの授業を行えないわけではなく、それぞれの教育委員会や学校で、カリキュラム・マネジメントを行って決めることになっている。その際、プログラミング教育を通して、「教科等で学ぶ知識及び技能等をより確実に身に付けさせること」が求められており、プログラミング自体の評価は行われず、その教科や単元の理解を評価することになる。

プログラミング教育の実施に向けた取り組み

以上のように、小学校のプログラミング教育を取り巻く状況は、さまざまな考えや立場を反映した結果、複雑化している。これに対して、現場の教員の不安を解消し、安心して取り組めることを目的に、文部科学省は「小学校プログラミング教育の手引」を作成している^{☆6}。

この手引では、プログラミングに関する学習活動を以下のように分類し、それぞれについて、目的や授業例を示している。

- A 学習指導要領に例示されている単元等で実施するもの
- B 学習指導要領に例示されていないが、学習指導要領に示される各教科等の内容を指導する中で実施するもの
- C 各学校の裁量により実施するもの(A、B及びD以外で、教育課程内で実施するもの)
- D クラブ活動など、特定の児童を対象として、教育課程内で実施するもの
- E 学校を会場とするが、教育課程外のもの
- F 学校外でのプログラミングの学習機会

合わせて、文部科学省、総務省、経済産業省を中心に、各地の教育委員会や企業、NPOなどが集まった「未来の学びコンソーシアム」が組織され、授業の事例集などを掲載した「小学校を中心としたプログラミング教育ポータル」を運営している^{☆7}。小学校プログラミング教育の手引で挙げられた授業例の一部は、こちらに実践例が掲載されている。

A 分類は教科書に載っている



図-2 新学習指導要領の例示(小学校プログラミング教育の手引(第三版), pp.26-27)

^{☆6} 小学校プログラミング教育の手引, https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm

^{☆7} 小学校を中心としたプログラミング教育ポータル, <https://miraino-manabi.jp/>

こともあり、ほとんどの小学校で行われると思われる。教科書会社もワークシートや教材などを用意しているので、教員も行いやすい。B分類は、積極的な取り組みを行っている一部の教育委員会や学校で行われる場合もあるだろう。とはいえ、その場合でも、ほかに道徳の教科化や英語が新しく入ったこともあり、十分な授業研究の時間が確保できないため、既存のポータルの例などをなぞる形になることが多いのではないだろうか。C分類は、授業時間の捻出が難しいことから行われな可能性が高い(モジュールと呼ばれる15分単位の隙間時間も使い切っていることもある)。D分類もごく一部の小学校で行われるだけと思われる。E分類以下は教育課程外なので小学校では除外としている。

移行期間中の授業

2018, 2019年度は、新学習指導要領への移行措置期間と位置づけられていた。しかし、その間の教育課程編成にプログラミングを含めなくてもよいことになっていたため、一部の研究指定校などで行われた授業例を一般化した形で紹介する。

低学年の生活科で児童が家遊びの計画を流れ図で作成する例(図-3)

冒頭で教員が行うことを説明した後に、4人から6

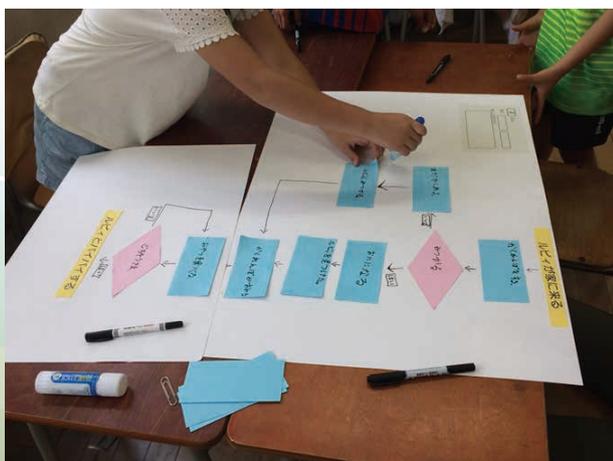


図-3 低学年の生活科で児童が家遊びの計画を流れ図で作成する例

人程度の班に分かれて、話し合いながら行われる。具体的には、各自が付箋紙に書いた活動を模造紙に貼り付け、その間を矢線で結ぶ。その際、いくつかの例外(雨のときはどうするか)などがあらかじめ示されることもある。記法は、おおむね JIS X 0121-1986 のフローチャートに準じたものを簡略化して使っている。

授業の最後に完成したものを各班が発表し、適宜児童と教員が質問やコメントを行う。最後に振り返りとして、何名かを指名して感想を発表させたり、ノートに書かせたりする。

学習の目当ては、教科のものとは別に、プログラミング的思考の育成とされることが多い。その際、順次・分岐・反復の理解が挙げられていることもある。

テーマが身近なこともあり、児童も積極的に参加しているように見受けられる。一方、完成した流れ図の妥当性の検証はほとんど行われな。極端なケースでは、菱形の分岐に活動が書かれていたりする。このような記法上の間違いや、論理的な間違いがあったとしても、それを教員や他の児童が指摘することは稀で、それらしく書かれていれば良しとしていることも多い。

そもそも、仕様に相当するものが定められていないので、教員の指導も児童の理解も地に足が着いていない、ふわふわした感じのまま終わってしまう。さらには、この流れ図に沿って実際にやってみたり、その結果をフィードバックして修正することもまずない(例外的に、体育のダンスの振り付けなどで行われることはある)。あくまで、一度限りの特別な活動と扱われている。

低学年の算数や国語の授業で、流れ図を使う行われる例(図-4)

これと似た授業を、算数で行うものに、2桁の引き算の筆算の手順を短冊を並べて流れ図で示すもの、国語で行うものに、ある言葉を辞書で引く方法を処理の箱で示すものなどがある。

これらのケースでは、行うべきこと(要求仕様)は明確であるが、それを詳細な手順に落とし込む際に曖昧さが残る問題がある。たとえば、辞書引きの手順化

に取り組むとき、粒度の大きな処理（「五十音表を調べる」など）を小さな処理に分割していないため、その手順を外化せず、頭の中で行ってしまう（分割統治や段階的詳細化の発想はまだない）。そのため、この流れ図に従って作業を行おうとしても再現できないことがある。その場合であってもそれを追求せず、なんとなくできたことにしてしまう。

筆算や、辞書引きでこの作業を行った後、時間を決めて計算ドリルを解いた数や、言葉を調べた数を競わせたりすることもある。その様子を観察して気付くのは、せっかく作った流れ図や短冊を参照しながら行う子がほとんどいないことである。そうだとすれば、この子たちにとって、この授業はなんだったのかという疑問が残る。

ある授業の最後で、担当教員は「コンピュータはこのやり方で処理を行っている」とまとめた。もちろん、計算機はこのような方法で処理を行っていない。この授業で行ったのは、人間が処理を進める手順を大雑把に表しただけである。

アンプラグドとCS アンプラグド

このような授業を、小学校では「アンプラグド」と呼んでいる。アンプラグドとは、コンセントに差していない、すなわち電気を使っていないということで、コンピュータを使っていないことを意味する。この言葉は、Tim Bellらが開発したCS（コンピュータサイエンス）アンプラグドから来ているが³⁾、計算機科学としての側面はほとんど失われており、教員が慣れ親しんだ従来の教授法に、フローチャートや、順次・反復・分岐などの用語を加えて、プログラミングの思考に合わせたものとして扱われている。

真剣にCS アンプラグドに取り組むとすれば、教える側にも相応の知識や能力が要求される。単にコンピュータを使っていないからアンプラグドと呼ぶことには問題があるだろう。また、このような授業が、総則にある「児童がプログラミングを体験しながら、コンピュータに意図した処理を行わせるために必要な論理的思考力を身に付けるための学習活動」と言えるかにも疑問がある。し

かし、前述の「コーディングを覚えることが目的ではない」ことがアンプラグドを行う根拠となっている。

さらに言えば、「教科等で学ぶ知識及び技能等」が従来に比べてより確実に身に付いたとも思えない。授業が終わった後の研究協議会の議論を聞いても、教員自身が納得しないまま行っていることがうかがえる。

では、なぜこのような歪な形になっているのだろうか。考えられるのは、教員がプログラミングの経験も知識もなく、授業でそれを行う意味を見出せていないことである。文部科学省の教職課程コアカリキュラムには、各教科の指導法に「情報機器及び教材の活用を含む」とあるだけで、直接にプログラミングは含まれておらず、これを受けた2019年度からの新教職課程にも入らなかった^{☆8}。したがって、大学でプログラミングを行った経験がある人、さらには、その教授法や効果について学んだ人の方が稀だと思われる。これは、中高年の教員だけでなく、若い初任者も同じである。

プログラミング恐怖症を超えて

筆者が教員研修の依頼を受けるとき、座学のみで演習は予定していないと言われることがある。演習を行う場合でも、座学の比率を増やしてほしいと要望されることもある。演習の際も、積極的にパソコンに触ろうとしない教員もいる。

☆8 改正前後の教職課程の科目等一覧：文部科学省，https://www.mext.go.jp/a_menu/shotou/kyoin/1414533.htm

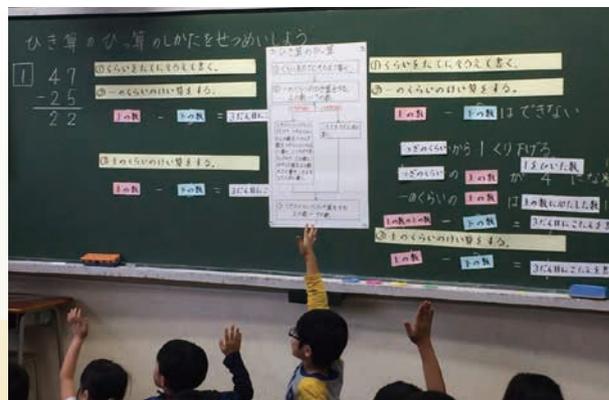


図-4 低学年の算数や国語の授業で、流れ図を使って行われる例

かつて、Seymour Papert は、その著書の中で、「数学恐怖症 (math phobia)」が、教育を通して子どもたちのの中に生まれていること、それによって子どもたちの数学に対する関心や理解が阻害されていることを指摘した⁴⁾。それと似たことが、プログラミングについても起こっており、「プログラミング恐怖症 (programming phobia)」と言えるような症状が教員の中にあると感ぜられる。

これを解決するには、児童がプログラミングすることの価値を教員に理解してもらう必要がある。

2014年から2016年にかけて、品川区立京陽小学校でプログラミング教育の実践が行われた⁵⁾。この実践の特徴は、全学年の全児童を対象にしたこと、1人1台のパソコンを実現したこと、プログラミングを独立させるのではなく、国語や算数など教科の授業として行ったこと、児童同士の学び合い (ペアプログラミングなど) と個々の進捗や理解度に応じた学習 (自発的な学習履歴の記録など) を重視したことである。

この実践の成果として、教員は「各教科のねらいをプログラミング学習と一体化させることにより児童の学習意欲が高まり、その結果、課題に対し粘り強く取り組んだり、話し合っ解決したりする姿が多く見られるようになった」こと、つまり、児童の学習態度の変容を挙げている。

もちろん、いきなりこうなったわけではなく、開始した当初はコンピュータに苦手意識を持つ教員も多く、研修もやられている感が否めなかった。しかし、プログラミングを通して児童が変わっていく状況を目の当

たりにして、教員も変容していった。

授業で使うプログラムも、教材会社が作った理解できない複雑なものを無理に使うのではなく、教員自身が作って理解できたものを教科や単元の中でどのように使うかを考えるようになった。

このような児童の変容を見るのに最も適しているのは、手引にあるC分類である。そこには、「プログラミングの楽しさや面白さ、達成感などを味わえる題材などでプログラミングを体験する取組」などが例示されている。

この授業を行う際には、教員が授業を統制するのではなく、なるべく児童に自由に弄り回させること (ティンカリング) が求められる。そうすれば、児童が、目の前で起こっていることを観察し、仮説を立て、コンピュータや友だちと対話し、試して、失敗し、そのことから学んでいる様子を観察できるだろう (図-5)。これは、新学習指導要領が提示する「主体的・対話的で深い学び」そのものである。

Resnick は、「子どもたちは彼らにとって個人的に意味のあるものを組み立てているときのみ、それを知的に行っているということである」と語っている。児童が創造的な学習者になることは、無理なアンブラグドでプログラミング的思考を身につけるよりも大切なことに違いない。Kayが15年前に指摘したように、そこに導く (ファシリテートする) のが学校や教員の役割である。

参考文献

- 1) 稲垣卓弥, 阿部和広, 山崎謙介, 横川耕二: 「教具」としての Squeak eToys とその小学校算数教育への適用, 研究報告コンピュータと教育 (CE), 15 (2009-CE-98), pp.57-63 (Feb. 2009).
- 2) Wing, J. M., 翻訳 中島秀之: Computational Thinking 計算論的思考, 情報処理, Vol.56, No.6, pp.584-587 (June 2015).
- 3) Bell, T. et al., 翻訳 兼宗 進: コンピュータを使わない情報教育アンブラグドコンピュータサイエンス, イーテキスト研究所 (Sep. 2007).
- 4) Papert, S., 翻訳 奥村貴世子: マインドストーム—子供, コンピュータ, そして強力なアイデア, 未来社 (Nov. 1982).
- 5) 久野 靖, 阿部和広, et al.: 学校まるごとわくわくプログラミング—品川区立京陽小学校の事例一, 情報処理, Vol.57, No.12, pp.1216-1238 (Dec. 2016).

(2020年5月25日受付)

